KLASIFIKASI *TWEETS* PADA TWITTER MENGGUNAKAN METODE *K-NEAREST NEIGHBOUR* (K-NN) DENGAN PEMBOBOTAN TF-IDF

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rakhman Halim Satrio
NIM: 125150200111115

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

KLASIFIKASI *TWEETS* PADA TWITTER MENGGUNAKAN METODE *K-NEAREST NEIGHBOUR* (K-NN) DENGAN PEMBOBOTAN TF-IDF

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rakhman Halim Satrio
NIM: 125150200111115

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing 2
M. Ali Fauzi, S.Kom., M.Kom.	Indriati, S.T., M.Kom.
NIK: 201502 890101 1 001	NIP: 19831013 201504 2 002

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 8 Juli 2019

Rakhman Halim Satrio

NIM: 125150200111115

PRAKATA

Segala puji syukur dipanjatkan kehadirat Allah SWT atas rahmat, hidayat dan pertolongan-Nya, sehingga penulis dapat menyelesaikan karya ilmiah berupa skripsi yang berjudul "Klasifikasi Tweets Pada Twitter Menggunakan Metode K-Nearest Neighbour (K-NN) Dengan Pembobotan TF-IDF".

Skripsi ini diajukan sebagai tugas akhir untuk memenuhi syarat untuk mendapatkan gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universitas Brawijaya. Penulis menyadari bahwa penyusunan skripsi ini tidak akan terwujud tanpa adanya bantuan, dukungan dan bimbingan dari berbagai pihak. Oleh sebab itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D., selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Agus Wahyu Widodo, S.T., M.Cs., selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak M. Ali Fauzi, S.Kom., M.Kom., dan Ibu Indriati, S.T., M.Kom., selaku dosen pembimbing yang dengan baik dan sabar memberikan arahan serta perhatiannya dalam penyelesaian skripsi ini.
5. Kedua orang tua penulis, Bapak Supriyadi dan Ibu Eni Lisetyati atas doa yang tidak henti-hentinya dipanjatkan untuk kelancaran dan kesuksesan penulis, serta adik kandung Putri atas dukungannya termasuk selama penulisan skripsi ini.
6. Bapak dan Ibu Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu yang bermanfaat selama penulis menempuh perkuliahan.
7. Sahabat pada komunitas yaitu Puma dan UB48 atas semangat, saran, dan motivasinya dalam pengerjaan skripsi ini.
8. Sahabat-sahabat penulis yaitu Agung Kharisma Sukarno, Candra Robiansyah, M. Rendra Husein Roisdiansyah, Joda Pahlawan Romadhona Tanjung, Dani Devito atas bantuan, semangat, saran, doa dan waktunya.
9. Semua pihak yang telah memberi semangat dan mendoakan penulis dalam penyelesaian skripsi ini yang tidak bisa disebutkan satu persatu.

Demikian skripsi ini dibuat, penulis menyadari bahwa masih terdapat banyak kekurangan sehingga kritik serta saran yang bersifat konstruktif akan berguna sekali untuk kesempurnaan skripsi ini. Semoga skripsi ini bisa bermanfaat serta dapat memberikan sumbangan berarti bagi pihak yang membutuhkan.

Malang, 8 Juli 2019

Penulis

rioirja8@yahoo.com

ABSTRAK

Rakhman Halim Satrio, Klasifikasi *Tweets* Pada Twitter Menggunakan Metode *K-Nearest Neighbour* (K-NN) Dengan Pembobotan TF-IDF.

Pembimbing: Mochammad Ali Fauzi, S.Kom., M.Kom. dan Indriati, S.T, M.Kom.

Twitter merupakan mikroblog yang sedang digemari oleh banyak orang dan berubah menjadi penyebar informasi yang sangat cepat saat ini. Informasi yang dihasilkan dan beredar melalui media ini sangat bebas dan beragam seperti berita, pertanyaan, opini, komentar, kritik baik yang bersifat positif maupun negatif. Klasifikasi merupakan semacam proses pada penambangan teks yang menggolongkan konten tertentu mengacu pada kesamaan skripnya. Dengan proses ini mengizinkan *tweets* tertentu yang berada pada Twitter digolongkan jadi satu bersumber pada kategorinya. Misalkan, berita sepakbola, voli, dan sepak takraw tergolong pada kategori olahraga. Proses pada klasifikasi diawali dengan *preprocessing*, dilanjutkan dengan pembobotan kata, kemudian kategorisasi yang terdiri dari penghitungan *cosine similarity.*

Preprocessing sendiri terdiri dari beberapa tahap yaitu pembersihan dokumen*, tokenizing*, *stopword removal*, dan *stemming*. Metode pembobotan kata yang digunakan pada skripsi ini adalah *Term Frequency–Inverse Document Frequency* (TF-IDF) dan menggunakan *K-Nearest Neighbor* (K-NN) sebagai metode klasifikasinya. Metode K-NN merupakan klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumya. Kategori yang digunakan diantaranya ekonomi, kesehatan, olahraga, otomotif dan teknologi.

Pengujian akurasi dari klasifikasi *tweets* pada Twitter dengan menggunakan metode *K-Nearest Neighbor* (K-NN) menghasilkan akurasi dimana total data berjumlah 140, dengan uraian 100 data latih dan 40 data uji serta nilai *k* yang dimasukkan adalah 1, 3, 5, dan 7, masing-masing hasilnya k = 1, akurasi sebesar 75,0%; k = 3, akurasi sebesar 72,5%; k = 5, akurasi sebesar 62,5%; k = 7, akurasi sebesar 55,0%.

Kata kunci: *K-Nearest Neighbor* (KNN), *Text Mining*, *Preprocessing*

ABSTRACT

Rakhman Halim Satrio, *Tweets* Classification In Twitter Using *K-Nearest Neighbour* (K-NN) Method With TF-IDF Weighting.

Supervisors: Mochammad Ali Fauzi, S.Kom., M.Kom. and Indriati, S.T, M.Kom.

Twitter is a microblog that is currently favored by many people and has turned out to be a very fast spreader of information at this time. Information generated and circulated through this media is very free and diverse, such as news, questions, opinions, comments, criticisms both positive and negative. Classification is a technique in text mining that classifies a content based on the similarity of the text. With this classification allows a tweets on Twitter to be grouped into one based on the category. For example, football, basketball and chess content are grouped into sports categories. The process of classification begins with preprocessing, followed by weighting words, then categorization which consists of calculating cosine similarity.

Preprocessing itself consists of several phases, that is document cleaning, tokenizing, stopword removal, and stemming. The word weighting method used in this thesis is Term Frequency-Inverse Document Frequency (TF-IDF) and uses K-Nearest Neighbor (K-NN) as its classification method. The KNN method is a classification of a set of data based on data learning that has been previously classified. Categories used include economics, health, sports, automotive and technology.

Accuracy testing of the classification of tweets on Twitter using the K-Nearest Neighbor (K-NN) method resulted in accuracy where the total data amounted to 140, with descriptions of 100 training data and 40 testing data and the values of k entered were 3, 5, and 8, each the result is when k = 1, the accuration is 75.0%; k = 3, accuration is 72.5%; k = 5, accuration is 62.5%; k = 7, accuration is 55.0%.

Keyword: *K-Nearest Neighbor* (KNN), *Text Mining*, *Preprocessing*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar belakang

Media sosial berupa *microblog* yang sekarang diminati khalayak luas salah satunya adalah *Twitter*, yang mana mengizinkan pemakainya membagikan serta melihat tulisan ringkas, dinamakan *tweets*. *Tweets* bisa dilihat dengan bebas, tapi juga bisa dikontrol sekadar boleh diketahui *user* tertentu yang mengikuti akun tersebut, dimana sebutannya dinamakan *follower*. Twitter berperan menjadi perantara penyiar kabar yang kencang bersamaan dengan meningkatnya *user* Twitter. Info yang disajikan serta bersirkulasi melewati media ini tidak dibatasi serta bermacam-macam layaknya kabar berita, kuesioner, pendapat, ulasan, kritikan bersifat membangun atau menjatuhkan (Putri, 2013). Pemakai Twitter bisa memposkan laporan yang diinginkan. Ketika pemakai mau membaca liputan yang berada pada laman depan Twitter, mereka menjumpai masalah, yakni tidak tersedianya kategorisasi *tweets*. *Tweets* yang muncul pada laman depan semua bergabung jadi satu, akibatnya pemakai yang mau membaca suatu *tweets* menjadi kebingungan untuk menemukannya. Contohnya, pemakai mau membaca *tweets* kabar mengenai bisnis, maka mereka perlu mencari satu demi satu *tweets* yang berisikan kabar bisnis.

Klasifikasi adalah semacam proses pada penambangan teks (text mining) yang menggolongkan konten tertentu mengacu pada kesamaan tulisannya. Dengan proses ini mengizinkan *tweets* tertentu yang berada pada Twitter digolongkan jadi satu bersumber pada kategorinya, semisal liputan sepakbola, basket, dan sepak takraw termasuk kedalam kategori olahraga. Dengan mengaplikasikan cara klasifikasi dalam *tweets* di Twitter bisa memudahkan pembaca sebab informasi yang berada pada Twitter digolongkan menurut kategorinya (Sriram et al., 2010). Pada dasarnya Twitter telah mengimplementasikan fitur klasifikasi *tweets* di situsnya yang mepermudah pemakai Twitter mendapat informasi berdasarkan pada kategorinya. Akan tetapi, ada suatu kekurangan pada pengimplementasian yang diterapkan Twitter yakni *tweets* yang dikelompokkan hanyalah sekadar akunnya. Lalu muncul persoalan dari hal itu, ialah pemakai biasa/tidak resmi sering bermasalah menentukan berita mana yang diikuti karena beritanya yang sering bercampur & berbeda topik (Phuvipadawat, 2010). Apabila diperoleh informasi dari akun yang tidak berhubungan akun lain itu, informasi itu akan memasuki kategori tertentu. Sebagai contoh, suatu akun resmi dari laman hiburan memposkan informasi pemilu, *tweets* yang diposkan akun tersebut tetap tergabung pada kategori informasi hiburan.

K-Nearest Neighbor (KNN) ialah salah satu prosedur pembelajaran mesin (*machine learning*) yang mengklasifikasi pada objek mengacu pada data pembelajaran dimana letak jarak adalah yang terdekat dari objek itu. Bersumber pada analisa yang dilakukan oleh penulis dengan memandang persoalan yang dilakukan peneliti lainnya pada penelitian klasifikasi twitter, maka diperlukan

penelitian untuk menyempurnakan penelitian yang telah ada agar klasifikasi menjadi lebih baik dan akurat lagi. Penulis mengerjakan observasi dengan judul **"Klasifikasi Tweets di Twitter dengan Menggunakan Metode K-Nearest Neighbour"**.

1.2 Rumusan permasalahan

Berlandaskan deskripsi yang terdapat di latar belakang, mampu dibuat rumusan permasalahan berikut ini.

1. Dengan cara apa mengaplikasikan metode klasifikasi *K-Nearest Neighbour* kedalam *tweets* pemakai Twitter?
2. Bagaimana cara membuat hasil menjadi semaksimal mungkin pada pengklasifikasian *tweets* dengan memakai *K-Nearest Neighbour*?
3. Bagaimanakah kualitas akurasi yang didapatkan dengan metode *K-Nearest Neighbour* pada klasifikasi?

1.3 Tujuan

Tujuan penelitian ini terbagi menjadi tujuan umum dan tujuan khusus. Diantaranya:

- Tujuan umum:

  Mengoptimalkan metode klasifikasi tweets pada Twitter dengan metode K-Nearest Neighbour

- Tujuan khusus:

  Mengelompokkan *tweets* pada Twitter dengan metode KNN.

1.4 Manfaat

Produk dari penelitian ini diharap bisa memberi kegunaan bagi pihak manapun. Manfaat dari penelitian ini ialah antara lain:

a. Bagi Penulis

   1. Mengaplikasikan ilmu yang telah didapatkan dari lingkup Jurusan Teknik Informatika Universitas Brawijaya ke dalam kehidupan bermasyarakat.
   2. Mendapatkan dan menambah pengetahuan berkenaan dengan metode klasifikasi.

b. Bagi Program Studi

   Menilai kesuksesan tahapan studi yang diberikan ke mahasiswa selama perkuliahan.

c. Bagi Fakultas

Menginformasikan hasil karya baru dari mahasiswa kepada masyarakat.

d. Bagi Twitter

1. Menambahkan fasilitas baru semacam pengelompokan informasi berlandaskan kategori *tweets* yang diposkan oleh pemakai.
2. Menjadi saran pada perusahaan guna pengembangan aplikasi supaya dapat menjadi lebih baik.

1.5 Batasan masalah

Agar tidak memperlebar wilayah pengkajian pada penelitian ini, maka penelitian ini diberi batas dalam perihal:

1. Dokumen yang dipergunakan ialah *tweets* Twitter dalam bahasa Indonesia.
2. Dokumen yang dipakai bersumber dari akun Twitter Detik & Kompas.
3. Prosedur pengelompokan yang dipergunakan yakni kaidah K-*Nearest Neighbour*.
4. Penggolongan dibedakan menjadi lima kategori, macam-macam kelasnya ialah kategori teknologi, kesehatan, olahraga, ekonomi, serta otomotif.

1.6 Sistematika pembahasan

Poin inti sistematika pengerjaan penulisan penelitian ini terdiri dari enam bab, yaitu:

BAB I **Pendahuluan**

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, serta sistematika penulisan.

BAB II **Landasan Kepustakaan**

Bab ini menjelaskan uraian dan penjabaran terkait teori, konsep, replika, serta metode yang berasal dari literatur ilmiah, yang berhubungan dengan tema permasalahan yang diangkat pada penelitian ini.

BAB III **Metodologi**

Bab ini menerangkan tentang langkah yang dipakai peneliti guna menyelesaikan penelitian dengan metode yang dipilih peneliti. Metode yang dipilih akan dijabarkan secara rinci mulai dari algoritme sampai alasan dipilihnya suatu metode.

BAB IV **Analisis dan Perancangan**

Bab ini menerangkan analisis kebutuhan, perancangan sistem, perancangan antarmuka dan juga perancangan uji coba serta evaluasi.

BAB V **Pembahasan**

Bab ini memberikan pembahasan yang rinci terkait hasil yang telah diperoleh dari pelaksanaan penelitian. Selain itu, di bagian ini akan menjawab pertanyaan atau masalah dalam penelitian.

BAB VI **Penutup**

Bab ini mengandung konklusi yang diperoleh dari penelitian yang telah dilaksanakan beserta masukan dalam rangka pengembangan penelitian kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pengkajian Terdahulu

Penelitian yang hendak dilaksanakan mengacu menurut pengkajian terdahulu, yakni pengkajian dimana dilaksanakan Perdana (2013). Dalam observasi tersebut dijelaskan bahwa pemakaian data latih dalam klasifikasi yang sumbernya yakni RSS (*Rich Site Summary*) sudah ditetapkan sebelumnya, yakni bidang olahraga, hiburan, berita, keuangan, otomotif, dan teknologi. Penelitian tersebut menunjukkan bahwa 79% dinyatakan benar berdasarkan kategori yang berhasil diklasifikasikan oleh sistem sebagai nilai rata-rata recall, sedangkan *precision* untuk mengukur banyaknya dokumen yang diklasifikasikan berlandaskan sistem serta dokumen tersebut diperoleh angka 80% dinyatakan benar. Sementara *F1 measure* untuk mengindikasikan gabungan antara nilai *precision* dan *recall* diperoleh angka 78% (Perdana, 2013). Namun demikian, capaian angka tersebut belum mencapai akurasi 90% sebagai capaian angka yang ideal. Hal tersebut diketahui dari *tweets* yang digunakan berupa *short-text*. Padahal seperti diketahui dari kajian yang lain bahwa *short-text* memang agak sukar dikelompokkan karena berbagai faktor, diantaranya *short-text* mempunyai beberapa ciri khusus yang harus menempel di setiap teksnya (Zelikovitz, 2005).

Penelitian lainnya yang terkait yakni penelitian yang dilaksanakan Kestrilia Rega Prilianti & Hendra Wijaya (2014) yang mengambil judul *"Aplikasi Text Mining untuk Automasi Penentuan Tren Topik Skripsi dengan Metode K-Means Clustering"* dengan memakai objek dokumen digital abstrak dari buku skripsi. Pada observasi tersebut memakai metode *Text Mining* yang berfungsi memecah kata-nya, kemudian dikelompokkan mengacu pada kedekatan antar dokumen. Ekspektasi dengan observasi ini yakni aplikasi dapat mencari secara mendalam dari topik-topik skripsi mahasiswa yang biasanya tergabung lewat *repository* digital perpustakaan Universitas. Dalam observasi ini percobaan dilaksanakan kepada dokumen skripsi dari enam program studi yang terdapat di Universitas Ma Chung, serta hasilnya didapatkan dari nilai *purity* sebesar 0.76 atau bisa dikatakan 76% berhasil di-*cluster* oleh sistem. *Purity* menghitung kemurnian dari suatu *cluster* yang direpresentasikan sebagai anggota *cluster* yang paling banyak sesuai di suatu kelas. Nilai *purity* yang mendekati 1 berarti semakin baik *cluster* yang didapatkan.

Pada penelitian lain dimana sebelumnya dikerjakan Qiang (2010) yang berjudul *"An Effective Algorithm for Improving the Performance of Naïve Bayes for Text Classification"*, menerangkan bahwasannya pemakaian metode *Naïve Bayes* yang tidak rumit & efektif dalam penggolongan teks namun seringkali tidak begitu sempurna kinerjanya.

Atas dasar itulah, peneliti mencoba untuk menyempurnakan hasil penelitian sebelumnya dengan menambahkan kategori *tweets* dimana bertujuan guna memperoleh hasil yang lebih akurat dari pengklasifikasian tersebut. Tidak jauh berbeda dengan observasi sebelumnya, kategori berita yang dipakai dalam

penelitian ini mencakup bidang teknologi, kesehatan, olahraga, berita ekonomi, dan otomotif.

2.2 Twitter

Twitter sebagai sarana jejaring social (*social networking*) yang juga sering disebut dengan istilah jaringan pertemanan merupakan bagian dari media sosial yang *trending* zaman sekarang. Sebagai layanan jejaring sosial, Twitter pada umumnya berbasis web dimana merupakan fasilitas internet, tujuannya diperuntukkan bagi komunitas *online* yang mempunyai kegiatan sejenis, kesamaan hobi, dan kesamaan pada aspek yang lain.

Twitter sebagai jejaring sosial memfokuskan pada layanan blog mikro (*microblogging*) dan RSS (Rich Site Summary) untuk penyampaian informasi, bahkan Twitter ini oleh sebagian pengguna diberi label SMS-nya internet. Meskipun sebagian orang menganggap *platform* dari Twitter memiliki keterbatasan dalam hal karakter, namun dalam hal yang lain justru Twitter mempunyai keunggulan dan keunikan dibandingkan dengan jejaring sosial atau yang lebih luas dibandingkan media sosial yang lain.

Keunggulan dan keunikan Twitter tersebut adalah: pertama, dari sisi tampilan antarmuka dan cara penggunaannya sangat sederhana (*simple*), sehingga Twitter menampilkan fitur-fitur yang efisien, *to the point*, seperlunya; kedua, karena informasi yang disampaikan secara *real time* berbasis waktu dan lokasi, sehingga Twitter memiliki daftar topik yang paling populer dan paling terkini yang dalam komunitas *online* disebut dengan *real time twitter trending topic*; ketiga, memiliki sifat yang terbuka, karena postingannya dapat diakses oleh siapapun, kapan pun, dan dimana pun, kecuali jika akun twitter tersebut diatur menjadi akun privat. Dengan demikian, pemakai Twitter bisa langsung berinteraksi tanpa harus menjadi *follower* dan tanpa harus konfirmasi pertemanan lebih dulu. Satu lagi yang dianggap sebagai keunggulan serta keunikan dari jejaring sosial ini adalah cuitan (*tweet*) yang di-*posting* memberikan peluang bagi pemakainya untuk dikenal lebih luas dengan memanfaatkan fitur *hashtag* oleh siapa pun yang sedang mencari *hashtag* yang sama.

Sebagaimana diketahui bahwa komunikasi melalui jejaring sosial yang relatif mendominasi dalam berbagi informasi antar penggunanya yang kicauannya disebut dengan *tweets* adalah Twitter yang memiliki kelebihan dengan 280 karakter. Pengguna Twitter bisa berbeda-beda tergantung dari *tweets* yang ada pada linimasa pada tiap pemakai atau penggunanya, sehingga pemakai Twitter berkenan mengikuti pemakai lainnya. *Tweets* yang terlihat hanyalah yang berasal dari *user* yang diikutinya (Perdana, 2013). Macam-macam fitur yang ada pada Twitter, diantaranya (Dixion, 2012):

1. *Followers & following*

   Followers (pengikut) adalah suatu akun/orang yang mengikuti akun lainnya, sedangkan *following* (mengikuti) adalah akun/orang yang diikuti suatu akun yang lain.

2. *Direct message*

   Twitter juga mengizinkan untuk mengirim pesan privat ke pengguna yang mengikuti akun tersebut.

3. *Twitter search*

   Salah satu fitur paling kuat di Twitter yaitu memberi kemudahan pemakai untuk mencari seseorang, kata kunci, subjek, dan tempat tertentu.

4. *Trending topics*

   Trending topics terdiri dari sepuluh topik yang banyak diposkan atau dibicarakan di Twitter dalam kurun waktu tertentu. *Trending topics* akan beredar dari berita, olahraga, hiburan yang menghibur, dan sebagainya.

2.3 Klasifikasi Teks

Seiring dengan kemajuan internet yang dari waktu ke waktu terus berkembang sangat cepat, berimplikasi pada kemudahan dan peningkatan kuantitas terutama dalam penyimpanan data ataupun dokumen. Apabila data atau dokumen tersebut tersimpan dalam jumlah yang sangat besar, misal sampai pada puluhan bahkan ratusan juta data atau dokumen, tentu akan sulit untuk melakukan klasifikasi data atau dokumen secara manual, karena pasti akan membutuhkan waktu yang cukup lama dan ketelitian yang amat tinggi, sementara di sisi lain ada kepentingan untuk segera dapat menemukan dan memanfaatkan data atau dokumen yang diperlukan. Salah satu teknik untuk melakukan klasifikasi tersebut adalah dengan cara *text mining*.

Klasifikasi teks ini ialah salah satu cara *text mining* yang dipakai memposisikan teks pada kelompok yang cocok dengan ciri-ciri teks itu yang didasarkan pada aturan yang sudah ditentukan. Berdasarkan pengelompokan atau klasifikasi teks ini, maka secara konseptual dapat memudahkan pemahaman cara penggolongan dokumen yang mempunyai peranan krusial pada kehidupan sesungguhnya (Sriram et al.,2010).

Pengkategorian teks pada hakekatnya dipakai untuk mengklasifikasikan dokumen yang lazim digunakan. Bahkan dalam kenyataannya tiap-tiap dokumen bisa saja tidak memiliki kategori atau sebaliknya mempunyai satu kategori atau bermacam-macam kategori. Dengan teknologi yang berkembang saat ini berupa *machine learning*, maka pengkajian tata cara klasifikasi dengan perumpamaan data latih, sanggup dipakai untuk mengklasifikasi kategori dokumen secara otomatis.

2.4 *Text Mining*

Dalam rangka menggali informasi yang diperlukan, *text mining* merupakan sarana bagi *user* berhubungan dengan sekelompok dokumen memakai *tools* (perlengkapan) analisis dengan menelusuri *data mining* yang salah satunya yaitu kategorisasi. Di samping itu, text mining sebagai sumber data dari himpunan teks

baik mempunyai pola semi terstruktur maupun yang tidak teratur juga dapat digunakan untuk memperoleh data yang bermanfaat dari sekelompok dokumen.

2.5 *Text Preprocessing*

Proses berikutnya adalah *text preprocessing*. *Text preprocessing* sebagai tahap persiapan data dilakukan karena banyaknya komentar dalam Twitter yang mengandung beragam jenis data, seperti: *hashtag*, *text*, *mention*, angka, *emoticon*, dan lain-lain yang menjadikan komentar tersebut memiliki tipe yang kompleks. Pada tahap ini data yang yang diperoleh dari penyiapan teks sebelumnya siap untuk diolah pada tingkatan selanjutnya. Ada beberapa *step* pada tahapan ini, yakni proses *tokenizing*, *filtering*, *stemming*, serta pembobotan *term* (Garcia, 2005).

2.5.1 *Tokenizing*

Sebagai *step* awal pada *text preprocessing,* operasi ini merupakan tahapan pemenggalan *string* input berlandaskan tiap kata yang menyusunnya (Sulhan, 2014). Tiap abjad *input* diganti menjadi abjad kecil. Seluruh tanda baca serta tanda hubung akan dihapuskan, juga seluruh karakter. Dengan kata lain, *tokenizing* secara teknis merupakan upaya untuk memecah sekelompok karakter dalam suatu teks ke dalam satuan kata, yakni proses membagi teks baik berupa kalimat, paragraf atau dokumen, menjadi bagian-bagian atau token-token tertentu.

Tokenizing merupakan proses pemisahan suatu rangkaian karakter berdasarkan spasi, bahkan pada waktu yang bersamaan bisa saja dilakukan penghapusan karakter tertentu, layaknya tanda baca. Di samping itu, *tokenizing* sebagai salah satu tahapan dari sistem dirancang untuk melakukan deteksi atas kemiripan.

2.5.2 Filtering

Step kedua setelah proses *tokenizing* adalah teknik *filtering,* yang fungsinya memperoleh kata-kata pokok dari hasil token (Sulhan, 2014). Dalam tahapan ini ditetapkan sebutan yang mewakili isi dokumen tersebut yang kemudian layak dipergunakan untuk menjelaskan maksud dokumen tersebut dan memisahkan dengan dokumen lainnya dalam koleksi. Pada tahapan ini pula dikerjakan penghilangan kosakata yang tidak bermanfaat atau dikenal dengan istilah *stoplist* atau istilah lainnya lagi *stopword*, yaitu daftar kata yang kerap dipakai, tetapi bukan mendeskripsikan konten dokumen, seperti kata ”yang”, “dan”, “di”, “dari” dan seterusnya. Namun sebaliknya menyisakan atau menyimpan kosakata yang dinilai penting, yang dikenal dengan istilah *wordlist*. *Stoplist* dikerjakan, karena kata-kata tersebut tak layak dipakai sebagai pembeda atau sebagai kata kunci dalam pencarian dokumen, sedangkan *wordlist* dipertahankan berfungsi sebagai kata kunci dalam pencarian dokumen. Dengan demikian, jumlah kata yang termasuk dalam *wordlist* bisa jadi akan lebih banyak dari pada *stoplist*.

2.5.3 Stemming

Step ketiga ialah *stemming*, yakni menemukan dasar kata dari tiap kata dari hasil tahap sebelumnya, yakni *filtering*. Pada tahap *stemming* ini, jika memakai bahasa Inggris, yang dihilangkan hanyalah *suffix*, namun dalam bahasa Indonesia, yang dihilangkan mencakup: *suffix*, *prefix*, *infix* serta *konfix* (Asian,2007).

Dari kondisi tersebut menunjukkan bahwa Algoritma *Stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya. Proses *stemming* pada teks berbahasa Indonesia lebih kompleks atau rumit, karena terdapat bermacam-macam imbuhan yang harus dibuang untuk mendapatkan akar kata-nya. *Stemming* digunakan untuk membatasi macam-macam bentuk kata yang berbeda menjadi bentuk dasarnya, dengan tujuan untuk meningkatkan kemampuan sistem dalam menemukan dokumen sesuai dengan query yang ada.

2.5.3.1 Stemming Nazief Andriani

Khusus untuk tahap *stemming* ini, untuk mencari akar kata atau dasar kata, dikenal sebuah teori, yaitu "Algoritme Porter" sebagai salah satu algoritme *stemming* yang kelebihannya adalah membutuhkan waktu lebih singkat dibandingkan dengan menggunakan Algoritma Nazief & Adriani, namun kelemahan dari proses *stemming* dengan memakai Algoritme Porter memiliki prosentase keakuratan atau presisi lebih rendah dibandingkan dengan *stemming* yang memakai Algoritme Nazief & Adriani. Begitu juga sebaliknya, Algoritma Nazief & Adriani sebagai salah satu algoritme *stemming* untuk teks berbahasa Indonesia, mempunyai keunggulan dalam hal persentase keakuratan atau presisinya lebih tinggi daripada algoritme lainnya. Pada umumnya algoritme ini amat diperlukan serta berpengaruh pada tahap *Information Retrieval* di dokumen berbahasa Indonesia. Di bawah ini diuraikan kata dasar dalam bahasa Indonesia yang tersusun oleh berbagai gabungan:

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Algoritme Nazief & Adriani yang diciptakan Bobby Nazief dan Mirna Adriani tersebut mempunyai prosedur seperti di bawah ini:

1. Awalnya carilah kata yang hendak di-stem pada kamus kata dasar. Bila diketemukan lalu diperkirakan kata merupakan dasar kata, kemudian algoritme berakhir.
2. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", maupun "-nya") dihapus. Bila berbentuk partikel ("-lah", "-kah", "-tah" ataupun "-pun"), langkah ini dilakukan kembali dengan tujuan membuang *Possesive Pronouns* ("-ku", "-mu", ataupun "-nya"), semisal ada.
3. Hilangkan *Derivation Suffixes* ("-i", "-an" maupun "-kan"). Apabila kata didapatkan pada kamus, algoritme selesai. Apabila tidak, lanjutkan ke tahap 3a

a. Bila “-an” sudah dihilangkan serta karakter terakhir kata tersebut ialah “-k”, jadi “-k” ikut dihilangkan. Bila kata itu didapatkan pada kamus, algoritme berakhir. Bila tak diketemukan, selanjutnya ke tahap 3b.
b. Sufiks yang dihilangkan (“-i”, “-an” ataupun “-kan”) dikembalikan, dilanjutkan tahap 4.

4. Hilangkan *Derivation Prefix*. Apabila ketika tahap ke-3 didapati sufiks yang dihilangkan, selanjutnya kerjakan tahap 4a, bila tidak, kerjakan tahap 4b.
    a. Cek tabel gabungan prefiks-sufiks yang tak diperbolehkan. Apabila didapati, algoritme akan berakhir, bila tidak, jalankan tahap 4b.
    b. For i = 1 to 3, tetapkan tipe prefiks, lalu hapuskan prefiks. Bila dasar kata belum didapatkan, jalankan tahap 5. Bila telah tuntas, proses algoritme selesai. Catatan: bila prefiks kedua sama dengan prefiks pertama, algoritme berakhir.
5. Melaksanakan *Recoding*.
6. Apabila seluruh prosedur sudah usai namun tidak sukses, maka kata awal dianggap dasar kata. Tahapan usai.

Tipe awalan dijelaskan melewati tahap di bawah ini:

1. Apabila awalannya yaitu: “di-”, “ke-”, ataupun “se-” lalu tipe awalannya ialah “di-”, “ke-”, atau “se-”.
2. Bila prefiksnya yaitu “te-”, “me-”, “be-”, maupun “pe-” lalu diperlukan operasi tambahan guna menetapkan jenis prefiksnya.
3. Apabila dua huruf awal selain “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, maupun “pe-” maka dihentikan.
4. Bila jenis awalan merupakan “none”, proses dihentikan. Seandainya jenis awalannya selain “none”, awalan bisa dilihat dalam Tabel 2.2. Hilangkan awalan bilamana ditemukan.

Tabel 2.1 Gabungan Awalan Akhiran Yang Tidak Diperbolehkan

Awalan	Akhiran yang tidak diperbolehkan
ke-	-i, -kan
me-	-an
di-	-an
se-	-i, -kan
be-	-i

Tabel 2.2 Prosedur Menetapkan Tipe Awalan bagi awalan "te-"

Following Characters				Tipe Awalan
Set 1	Set 2	Set 3	Set 4	
"-r-"	"-r-"	–	–	none
"-r-"		–	–	ter-luluh
"-r-"	not (vowel or "-r-")	"-er-"	vowel	ter
"-r-"	not (vowel or "-r-")	"-er-"	not vowel	ter-
"-r-"	not (vowel or "-r-")	not "-er-"	–	ter
not (vowel or "-r-")	"-er-"	vowel	–	none
not (vowel or "-r-")	"-er-"	not vowel	–	te

Tabel 2.3 Jenis Awalan Menurut Tipe Awalannya

Tipe Awalan	Awalan yang patut dihilangkan
ke-	ke-
di-	di-
te-	te-
se-	se-
ter-	ter-
ter-luluh	ter

Dalam rangka menanggulangi keterbatasan dalam algoritme tersebut, ditambahkanlah aturan-aturan di bawah ini:

1. Aturan kata yang diulang
    a. Bila ada dua kata yang ternyata adalah kata yang sama, dasar kata ialah bentuk tunggalnya, misalnya : “buku-buku” dasar kata-nya yaitu “buku”.
    b. Kata lain, contohnya “bolak-balik”, “berbalas-balasan, serta ”seolah-olah”. Untuk meperoleh dasar kata-nya, kedua kata diartikan secara terpisah. Bila keduanya mempunyai dasar kata yang sama, akan diubah jadi bentuk tunggal, misalnya: kata “berbalas-balasan”, “berbalas” dan “balasan” memiliki dasar kata yang sama yakni “balas”, maka root word “berbalas-balasan” ialah “balas”. Sebaliknya, dalam kata “bolak-balik”, “bolak” dan “balik” mempunyai dasar kata berbeda, lalu dasar kata-nya ialah “bolak-balik”.
2. Tambahan bentuk awalan & akhiran beserta ketentuannya.
    a. Tipe awalan “mem-“, kata yang didahului dengan awalan “memp-” mempunyai tipe awalan “mem-”.
    b. Tipe awalan “meng-“, kata yang didahului awalan “mengk-” mempunyai tipe awalan “meng-”.

Di bawah ini contoh-contoh petunjuk yang terdapat dalam awalan sebagai pembentuk kata dasar.

1. Awalan SE-

Se + seluruh konsonan dan vokal tetap tak berubah

Misal : a. Se + suap = sesuap
c. Se + buah = sebuah
b. Se + kelas = sekelas
d. Se + eksis = seeksis

2. Awalan ME-

Me + vokal (a,i,u,e,o) jadi sengau “meng”

Misal : a. Me + isi = mengisi
d. Me + elak = mengelak
b. Me + ambil = mengambil
e. Me + antre = mengantre
c. Me + ukir = mengukir

Me + konsonan b jadi “mem”

Misal : a. Me + baca = membaca
b. Me + bangun = membangun

Me + konsonan c jadi “men”

Misal : a. Me + contek = mencontek
b. Me + cabut = mencabut

Me + konsonan d jadi “men”

Misal : a. Me + doktrin = mendoktrin b. Me + daftar = mendaftar

Me + konsonan g dan h jadi “meng”

Misal : a. Me + gaet = menggaet b. Me + hardik = menghardik

Me + konsonan j jadi “men”

Misal : a. Me + jahit = menjahit b. Me + jauh = menjauh

Me + konsonan k jadi “meng” (luluh)

Misal : a. Me + kejar = mengejar b. Me + kontrol = mengontrol

Me + konsonan p jadi “mem” (luluh)

Misal : a. Me + pakai = memakai b. Me + parkir = memarkir

Me + konsonan s jadi “meny” (luluh)

Misal : a. Me + sadap = menyadap b. Me + sadur = menyadur

Me + konsonan t jadi “men” (luluh)

Misal : a. Me + tambah = menambah b. Me + tagih = menagih

Me + konsonan (l,m,n,r,w) jadi tetap “me”

Misal : a. Me + lenggang = melenggang d. Me + rangkul = merangkul

b. Me + merah = memerah e. Me + wabah = mewabah

c. Me + nikah = menikah

3. Awalan KE-

Ke + semua konsonan dan vokal tetap, tak berubah

Misal : a. Ke + balik = kebalik b. Ke + arah = kearah

4. Awalan PE-

Pe + konsonan (h,g,k) dan vokal jadi “per”

Misal : a. Pe + hias + an = perhiasan

b. Pe + gerak + an = pergerakan

c. Pe + kumpul + an = perkumpulan

Pe + konsonan “t” jadi “pen” (luluh)

Misal : a. Pe + tunda = penunda b. Pe + tabuh = penabuh

Pe + konsonan (j,d,c,z) jadi “pen”

Misal : a. Pe + jambret = penjambret c. Pe + cabut = pencabut

b. Pe + dekam = pendekam d. Pe + zina = penzina

Pe + konsonan (b,f,v) jadi “pem”

Misal : a. Pe + bawa = pembawa b. Pe + faktor = pemfaktor

Pe + konsonan “p” jadi “pem” (luluh)

Misal : a. Pe + pantau = pemantau b. Pe + penggal = pemenggal

Pe + konsonan “s” jadi “peny” (luluh)

Misal : a. Pe + simpan = penyimpan b. Pe + sapu = penyapu

Pe + konsonan (l,m,n,r,w,y) tetap tak berubah

Misal : a. Pe + laku = pelaku c. Pe + nilai = penilai

b. Pe + mudik = pemudik d. Pe + wayang = pewayang

2.5.4 Pembobotan

Step keempat pada *text preprocessing* adalah *term weighting,* yaitu sebuah klasifikasi dokumen dengan menggunakan metode *TF.IDF* (Frekuensi Term-Frekuensi Dokumen Invers), yakni suatu kaidah pembobotan paling lazim digunakan untuk mengilustrasikan sebuah dokumen dalam model ruang vektor (*vector space model*). Dalam klasifikasi teks, ada dua kaidah pembelajaran mesin dimana sering digunakan dengan metode *TF-IDF*, yakni *k*-NN serta SVM. *TF-IDF* biasanya dipakai guna mengkomparasikan kesamaan (*similarity*) vektor *query* terhadap vektor dokumen (Soucy & Mineau, 2005).

Term Frequency (*TF*) ialah elemen yang menetapkan bobot *term* pada dokumen tertentu mengikuti jumlah kemunculannya pada dokumen tersebut. Skor jumlah kemunculan kata tertentu (*term frequency*) ikut dipertimbangkan pada penyerahan bobot pada kata tertentu (*term frequency*). Makin tinggi jumlah kemunculan suatu kata atau *term* pada suatu dokumen, makin bertambah bobot kata tersebut dalam dokumen, dengan kata lain kata tersebut sangat sesuai menjelaskan dokumen tersebut.

Inverse Document Frequency (*IDF*) dilakukan untuk mengurangi dominasi kata yang kerap keluar dalam kumpulan dokumen. Dapat dikatakan bahwasannya kata yang banyak keluar dari kumpulan berbagai dokumen adalah kata umum (*common term*). Sehingga kata umum tersebut tidaklah terlalu penting. Sebaliknya, kata yang ditemukan pada dokumen yang sedikit harus dilihat sebagaimana kata yang kian penting (*uncommon term*) dibandingkan kata yang keluar pada kebanyakan dokumen. Dengan mencari faktor kejarangmunculan kata (*term scarcity*) di kumpulan koleksi dokumen, bobot yang diberikan ke dalam kata akan lebih sesuai. Untuk menentukan bobot tersebut, perlu dicari faktor keterbalikan frekuensi dokumen yang memuat kata tertentu (*inverse document frequency*), yang diusulkan oleh George Zipf. Beliau memperhatikan bahwasannya frekuensi dari sesuatu cenderung berkebalikan secara proporsional dengan susunannya (Zafikri, 2008).

TF-IDF bisa dirumuskan dalam rumus 2.1

$$w(t, d) = tf(t, d) * idf, \quad (2.1)$$

$$idf = log\left(\frac{N}{df}\right) \quad (2.2)$$

Keterangan:

tf(*t*, *d*) : kemunculan kata t pada dokumen d

N : jumlah dokumen pada kumpulan dokumen

df : jumlah dokumen yang mengandung *term t*.

Selanjutnya rumus untuk mengkalkulasi bobot kata (w_i) pada dokumen dikalkulasi memakai Rumus 2.3

$$w_i = TF(t_i, d) \times IDF(t_i) \quad (2.3)$$

Penjelasan:

w_i : bobot kata (*term*) pada dokumen d

$TF(t_i, d)$: jumlah kata (*term*) t_i yang ada di dokumen d

$IDF(t_i)$: *inverse document frequency* dari kata (*term*) t_i

2.6 *Vector Space Model*

Vector Space Model (VSM) umum dipakai dengan tujuan menganalisis kedekatan antara dua objek. Sederhananya, tiap-tiap objek diwujudkan berupa format vektor serta kedekatan keduanya dikalkulasi berlandaskan *vector processing*. VSM sering dipakai untuk mengukur kesamaan atau kemiripan (*similarity term*) antara suatu dokumen dan suatu query dengan cara pembobotan term.

Skema ini mengkalkulasi derajat *similarity* antara tiap dokumen yang disimpan pada sistem dengan *query* yang diinputkan pemakai. Pola ini awalnya ditunjukkan Salton (1989). Bobot *query* serta dokumen diwujudkan pada format vektor, layaknya Persamaan 2.4 berikut.

$$Q = \left(W_{q1}, W_{q1}, W_{q1}, \dots, W_{qt}\right) \text{ dan } D_i = \left(W_{i1}, W_{i1}, W_{i1}, \dots, W_{it}\right) \quad (2.4)$$

Keterangan:

Q : *Query*

D : Dokumen

W : Bobot

Selanjutnya dipakailah persamaan ternormalisasi guna menemukan besaran cosinus sudut diantara dua *vector* dari tiap bobot dokumen (WD) dan bobot *query* (WQ).

2.7 *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) ialah satu metode di antara metode lain yang paling simpel guna menyelesaikan permasalahan klasifikasi (Adeniyi, Wei, & Yongquan, 2016). Algoritme ini kerap dipakai dengan tujuan klasifikasi teks & data (Samuel, Delima, & Rachmat, 2014). Pada kaidah ini dilaksanakan klasifikasi pada obyek berlandaskan data, yang mana jaraknya terdekat terhadap obyek yang dimaksud (Hardiyanto & Rahutomo, 2016).

Jarak antara dua titik x_1 dan x_2 didefinisikan sebagai berikut.

$$d(x_1, x_2) = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + \cdots + \left(x_{1p} - x_{2p}\right)^2}$$

$$= \sqrt{\sum_{j=1}^{p}(x_{1j} - x_{2j})^2} \qquad (2.5)$$

Keterangan:

$d(x_1,x_2)$: jarak antara variabel x_1 dan x_2
x : variabel
p : jumlah dimensi variabel

Tahapan pengkategorian dokumen *testing* dengan memakai algoritme *K-NN* ialah:

1. Menetapkan kriteria nilai k (jumlah tetangga terdekat)
2. Mengkalkulasi kuadrat *euclidean distance* objek kepada data latih yang disediakan
3. Mengurutkan hasil no. 2 dengan *ascending* (urut mulai nilai yang besar menuju kecil)
4. Menggabungkan kategori Y (Klasifikasi *nearest neighbor* mengacu pada *k values*)
5. Dengan memakai kategori *nearest neighbour* paling mayoritas, bisa diperkirakan kategori objek tersebut

2.7.1 Cosine Similarity (CosSim)

Metode *Cosine Similarity* ialah kaidah yang berfungsi mengkalkulasi kesamaan antara dua dokumen tertentu.

Penetapan kecocokan dokumen dengan *query* diperhatikan sebagai pengukuran (*similarity measure*) antara *vector* dokumen (D) dengan *vector query* (Q). Makin serupa *vector* dokumen tertentu dengan *vector query,* maka dokumen bisa dilihat bertambah cocok dengan query (Putri, 2013). Persamaan yang dipakai dalam mengkalkulasi *cosine similarity* pada persamaan 2.10:

$$cosSim(X, d_j) = \frac{\sum_{i=1}^{m} x_i . d_{ji}}{\sqrt{(\sum_{i=1}^{m} x_i)^2} . \sqrt{(\sum_{i=1}^{m} d_{ji})^2}} \quad (2.6)$$

Keterangan:

X	: dokumen *testing*
d_j	: dokumen *training*
x_i & d_{ji}	: angka bobot yang diberi ke dalam tiap *term* kepada dokumen.

Dalam melakukan sebuah pengujian atas dokumen adakalanya dilihat dari kedekatan *query* dan dokumen yang ditandakan oleh sudut yang dibuat. Kecenderungan nilai *cosinus* yang besar, umumnya ditandai dengan dokumen yang cenderung sesuai *query*. Pada dasarnya perhitungan yang menggunakan rumus persamaan (2.10) dipakai untuk mengetahui angka kesamaan ketika melakukan tahap mengkomparasikan dokumen yang cocok terhadap dokumen yang sudah tersedia ataupun dokumen yang lain.

2.8 Akurasi

Akurasi sebagai salah satu indikator keberhasilan sebuah sistem memegang peranan yang sangat penting. Untuk melakukan pengukuran akurasi, dilakukan dengan cara mengukur kedekatan antara hasil pengukuran dengan angka sebenarnya. Persamaannya yaitu sebagai berikut.

$$akurasi = \frac{\sum data\ uji\ benar}{\sum jumlah\ h\ total\ data\ uji} \times 100\% \quad (2.7)$$

BAB 3 METODOLOGI

Sebagaimana layaknya sebuah penelitian sistem, maka tahapan dalam metode penelitian ini dilakukan melalui: metode secara umum, alur sistem, penghimpunan data, perencanaan sistem, penerapan sistem, pengujian sistem, analisis, terakhir adalah kesimpulan & saran.

3.1 Tipe Penelitian

Penelitian ini memakai tipe non-implementatif analitik. Tipe ini menitikberatkan kepada investigasi terhadap suatu keadaan yang kemudian membuahkan kajian ilmiah. Analitik berarti penelitian yang berusaha mendeskripsikan derajat relasi antar komponen pada objek penelitian dengan kondisi tertentu yang sedang diteliti.

3.2 Metode Umum

Metode umum menjelaskan teori yang dipakai guna mendukung penulisan skripsi. Teori-teori tersebut melingkupi:

a. Text Mining

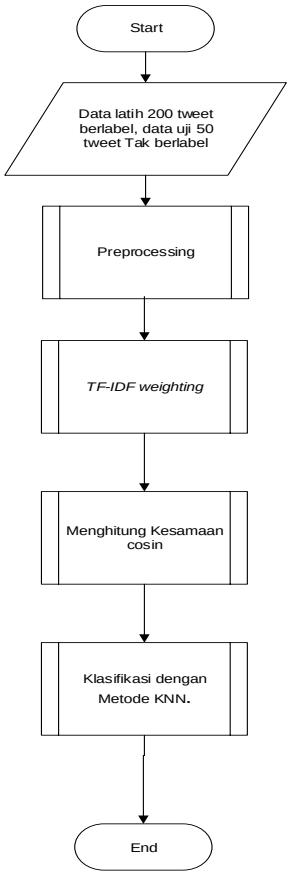
   Suatu prosedur penggalian informasi yang mana suatu pengguna berinteraksi dengan sekelompok dokumen menggunakan *tools* (perlengkapan) analisis yang merupakan komponen yang banyak pada *data mining* dimana salah satunya ialah kategorisasi.

b. Metode Klasifikasi KNN

   Algoritme ini melaksanakan klasifikasi pada obyek mengacu kepada data yang berjarak amat dekat dengan obyek itu.

3.3 Alur Sistem

Pada tahap ini diberikan diagram sistem untuk menjelaskan langkah-langkah penelitian yang akan dikerjakan.



Gambar 3.1 Alur sistem secara umum

3.4 Lokasi Penelitian

Penelitian ini dilaksanakan di Fakultas Ilmu Komputer (FILKOM), tepatnya di laboratorium riset FILKOM. Di dalamnya ada 8 grup riset di bidang Ilmu Komputer. Penelitian dilakukan pada Grup Riset Sistem Cerdas.

3.5 Pengumpulan Data

Penelitian ini menggunakan data sekunder. Data yang diperlukan meliputi *dataset* serta data latih dan data uji. *Dataset* diambil dari *tweets* konten berita di portal berita detik.com & kompas.com yang mencakup 140 (Seratus Empat Puluh) berita dengan maksud untuk memperoleh kosakata pada *term*, sedangkan data latih dan data uji diambil dari *tweets Dataset* Twitter.

3.6 Perancangan Sistem

Untuk menyusun perancangan sistem, harus dilaksanakan lebih dulu analisa kebutuhan serta telah diperoleh data yang diperlukan. Adapun tahapan yang dipersiapkan untuk membuat sistem seperti yang diuraikan di bawah ini.

1. Perancangan Arsitektur

   Tahap pertama yang perlu dilakukan adalah menyusun perancangan jalannya sistem, dengan cara melakukan proses *text processing* yang dilanjutkan dengan melakukan klasifikasi dengan menggunakan metode KNN.

2. Perancangan Data

   Setelah tahap perancangan arsitektur, langkah berikutnya adalah penyusunan perancangan database dalam rangka penyimpanan data dalam sistem yang disiapkan. Seperti yang diuraikan pada bagian sebelumnya, dataset yang digunakan bersumber dari *tweets* akun detik.com & kompas.com, sementara data *training* & *testing* diambil dari dataset dalam Twitter.

3. Perancangan Antarmuka

   Tahapan akhir dari proses perancangan sistem ini, disusun perancangan antarmuka dari sistem yang disiapkan. Antarmuka ini digunakan sebagai sarana komunikasi antara *user* dengan sistem.

3.7 Peralatan Pendukung

Peralatan pendukung menjelaskan tentang spesifikasi piranti keras & piranti lunak yang dipergunakan pada penelitian.

1. *Software* yang dipakai untuk mendukung penelitian ini diantaranya adalah:
   a. *Operating System* Microsoft Windows 7/8/10
   b. *Database Management System* (DBMS) MySQL atau SQLite Manager
   c. *Web Browser* Google Chrome atau Mozilla Firefox
   d. *Editor* Bahasa Pemrograman Netbeans 8.0

2. Perangkat keras (*Hardware*) yang dipakai pada pelaksanaan penelitian ini yaitu Komputer (PC)/Laptop, spesifikasinya RAM 2,00 GB, monitor 14”, Harddisk 750 GB.

3.8 Pengujian dan Analisis

Sebagaimana dikutip pada uraian pengumpulan data di atas, bahwa pengujian sistem dilakukan dari 140 (Seratus Empat Puluh) berita dan 140 (Seratus Empat Puluh) *tweets*, yang hasilnya dianalisis untuk mengetahui dan mengevaluasi dari sistem yang telah disiapkan.

Skema pengujian yang dilaksanakan yakni seperti di bawah ini:

1. Dari nilai *cosine similarity*, dilakukan pencarian nilai *threshold* terbaik dengan mencoba memilih lima sampel *threshold* secara acak.
2. Memeriksa hasil akurasi yang didapat dengan cara melakukan proses klasifikasi menggunakan pembobotan Tf.Idf dan metode *K-Nearest Neighbour*.

Setelah dilakukan pengujian, akan didapatkan hasil yang diberikan oleh sistem, yang selanjutnya dapat dipakai sebagai bahan analisa dan pengambilan kesimpulan.

3.9 Kesimpulan dan Saran

Tahapan akhir dari seluruh tahapan penelitian ini adalah pengambilan kesimpulan dan saran. Kesimpulan disusun berlandaskan uraian dan data yang telah dikaji dan dianalisis, sekaligus merupakan jawaban rumusan masalah penelitian ini. Sedangkan saran dimaksudkan sebagai rekomendasi atas permasalahan yang belum terpecahkan dalam penyusunan sebuah sistem, sekaligus rekomendasi untuk kajian atau penelitian lanjutan.

BAB 4 ANALISIS DAN PERANCANGAN

Analisis kebutuhan & perancangan terhadap sistem "Klasifikasi *Tweets* Pada Twitter Dengan Menggunakan Metode *K-Nearest Neighbour* (*K-NN*)" menjadi materi inti dalam bab ini. Materi yang disajikan pada bab ini meliputi tiga hal, yaitu pertama, analisa kebutuhan sistem; kedua, perancangan *software*; dan ketiga adalah perancangan *interface*.

4.1 Analisis Kebutuhan Sistem

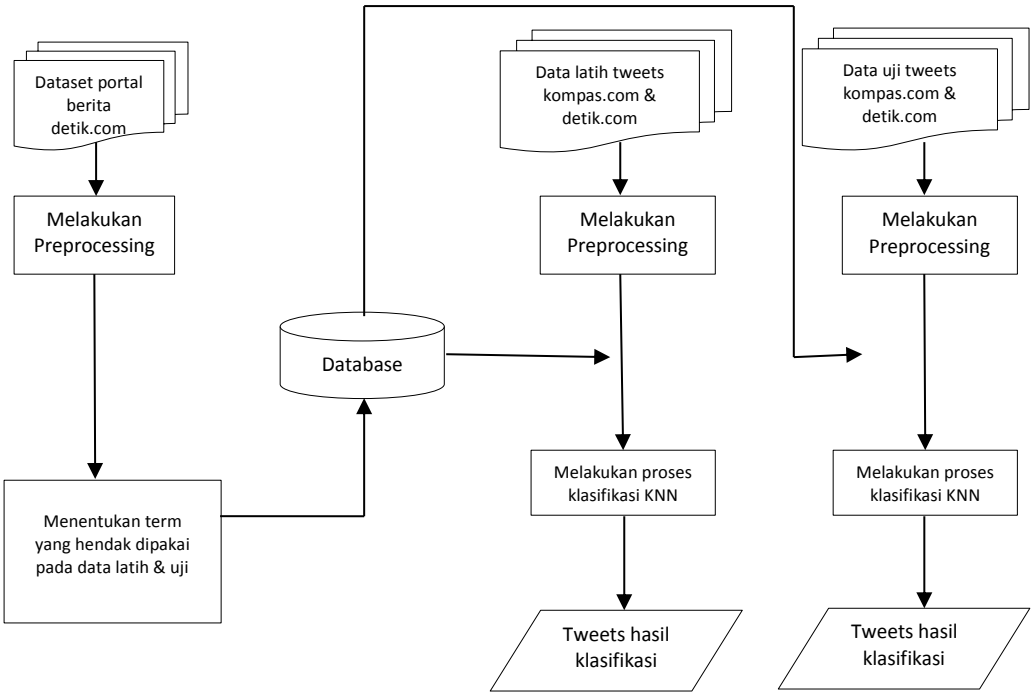
Penggunaan analisis kebutuhan sistem dalam rangka guna mengetahui apa saja yang harus disiapkan ketika pembuatan aplikasi tersebut dilakukan. Perangkat yang harus disiapkan diantaranya yaitu pertama, *hardware* berupa komputer PC, spesifikasi RAM 4,00 GB & monitor 14"; kedua, *software* berupa OS Microsoft Windows 7, aplikasi Netbeans 8.0 serta bahasa *programming* Java; dan yang ketiga yakni kebutuhan data bisa dalam portal berita detik.com maupun data *tweet* detik dan kompas.

4.2 Perancangan Perangkat Lunak

Dalam sistem klasifikasi *tweet* pada media sosial Twitter, paling tidak ada 4 (empat) hal yang harus dipersiapkan dalam perancangan perangkat lunak, yaitu pertama, runtutan kerja sistem secara lazim; kedua, perencanaan sistem manajemen data; ketiga adalah perencanaan rincian tata cara kerja sistem (*flowchart*); dan yang keempat adalah *prototype* gambaran aplikasi atau bisa dinamakan *interface*.

4.2.1 Diagram Blok Alur Kerja Sistem

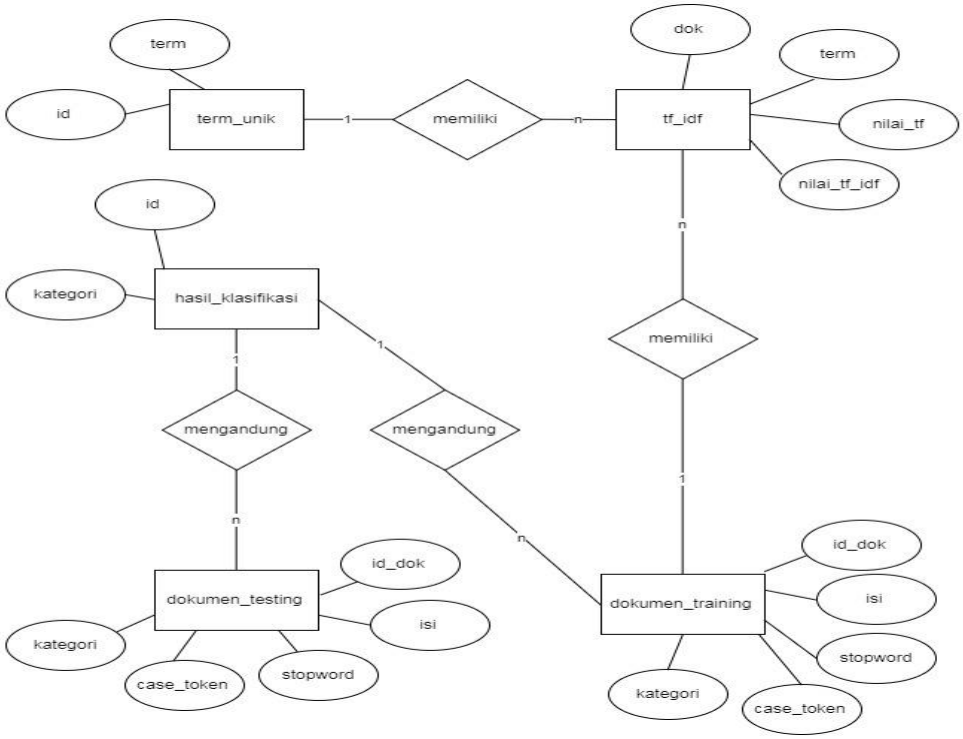
Berikutnya adalah penjelasan mengenai cara kerja sistem seperti yang ditampilkan dalam Gambar 4.1. Dari cara kerja sistem inilah akan dihasilkan klasifikasi *tweets*. Untuk mengklasifikasi cara kerja sistem ini, digunakan metode KNN. Sebelum ditampilkan dalam dataset, data training, dan data testing, maka sistem akan bekerja lebih dulu melalui *preprocessing* untuk menghapus kata atau bahasa yang dinilai tidak bermanfaat. Dataset yang digunakan dapat bersumber dari situs web berita detik.com, sementara untuk data training & data testing akan diambil melalui tweets dari kompas.com dan detik.com.



Gambar 4.1 Diagram utama Sistem Klasifikasi KNN

4.2.2 Perancangan Sistem Manajemen Data

Dalam skema berikut ini ditampilkan Entity-Relationship Diagram (ERD) untuk klasifikasi *KNN* sebagai aplikasi dari perancangan sistem manajemen data.



Gambar 4.2 ERD Klasifikasi KNN

Dalam tampilan ERD tersebut dipahami bahwasannya dalam perencanaan replika manajemen data terdapat sejumlah entitas dimana nantinya menyimpan data guna pemrosesan data dokumen. Beberapa entitas itu diantaranya:

1. Dokumen Training

Entitas berikut menyimpan sejumlah dokumen yang diambil dari *tweets* situs portal berita Detik (detik.com) & Kompas (kompas.com) yang dipakai dalam tahapan klasifikasi. Di bawah ini merupakan tipe data yang dipunyai oleh setiap atribut:

Tabel 4.1 Tabel keterangan Dokumen Latih

Nama Atribut	Tipe Data	Kegunaan
id_dok	INTEGER(12)/ PRIMARY KEY	fungsinya menetapkan runtunan penyimpanan dokumen sebelum pemrosesan
isi	VARCHAR(10010)	menyimpan konten dokumen berita yang hendak dikerjakan pemrosesan
case_token	VARCHAR(10010)	menyimpan konten dokumen berita yang telah dilaksanakan *case folding* & *tokenizing*
stopword	VARCHAR(10010)	menyimpan konten dokumen berita yang telah diterapkan *filtering* ataupun *stopword removal*

2. Dokumen Testing

Fungsi dari dokumen testing pada dasarnya adalah menyimpan sejumlah dokumen berita *tweets* dari sumber sama seperti di atas yang telah dipakai pada waktu proses klasifikasi. Tabel di bawah ini merupakan tipe data dari tiap-tiap atribut.

Tabel 4.2 Tabel keterangan Dokumen Uji

Nama Atribut	Tipe Data	Kegunaan
id_dok	INTEGER(12)/ PRIMARY KEY	menetapkan runtunan penyimpanan dokumen sebelum dikerjakan pemrosesan
Isi	VARCHAR(10010)	menyimpan konten dokumen berita yang hendak diproses
case_token	VARCHAR(10010)	menyimpan konten dokumen berita dimana telah diterapkan *case folding* serta *tokenizing*

stopword	VARCHAR(10010)	menyimpan konten dokumen berita dimana telah diaplikasikan *filtering* ataupun *stopword removal*

3. Term Unik

Fungsi dari entitas ini, menyimpan *term* unik yang didapatkan dalam seluruh dokumen training serta testing. Tabel di bawah ini merupakan tipe data dari tiap-tiap atribut.

Tabel 4.3 Tabel keterangan Term Unik

Nama Atribut	Tipe Data	Kegunaan
id	INTEGER(100)/ PRIMARY KEY	fungsinya menetapkan urutan penyimpanan *term* unik
term	VARCHAR(260)	mengandung kumpulan *term* unik dari seluruh dokumen yang dipakai sebagai set data

4. Hasil Klasifikasi

Fungsi dari entitas ini ialah menyimpan kategori yang dipakai dalam langkah klasifikasi. Tabel di bawah ini merupakan tipe data dari tiap atribut.

Tabel 4.4 Tabel keterangan Hasil Klasifikasi

Nama Atribut	Tipe Data	Kegunaan
Id	INTEGER(110)/ PRIMARY KEY	gunanya menetapkan urutan penyimpanan kategori tertentu
kategori	VARCHAR(260)	memuat kategori yang dipakai dalam tahapan klasifikasi

5. TF-IDF

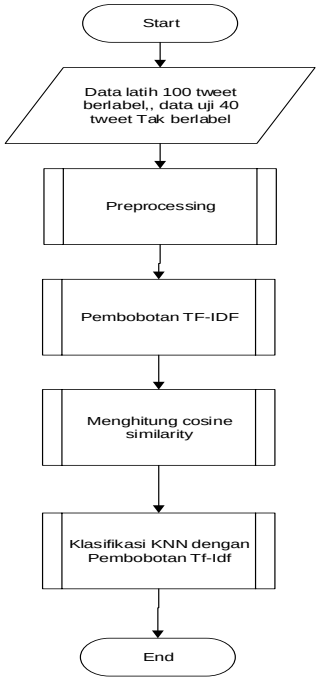
Fungsi dari entitas ini adalah menyimpan nilai TF serta besaran TF-IDF dalam mengkalkulasi nilai pembobotan. Tabel di bawah ini merupakan tipe data dari tiap-tiap atribut.

Tabel 4.5 Tabel keterangan TF-IDF

Nama Atribut	Tipe Data	Kegunaan
term	VARCHAR(260)/ PRIMARY KEY	gunanya menetapkan runtunan penyimpanan kategori tertentu
dok	INTEGER(12)	fungsinya yaitu menetapkan runtunan penyimpanan dokumen
nilai_tf	DOUBLE(110)	gunanya yakni menyimpan angka tf dari tiap-tiap *term*
nilai_tf_idf	DOUBLE(110)	fungsinya yaitu menyimpan angka tf-idf dari hasil *term weighting*

4.2.3 Diagram Alir Sistem

Di bawah ini merupakan *flowchart* Klasifikasi *Tweets* Pada Twitter, digambarkan oleh Gambar 4.6.



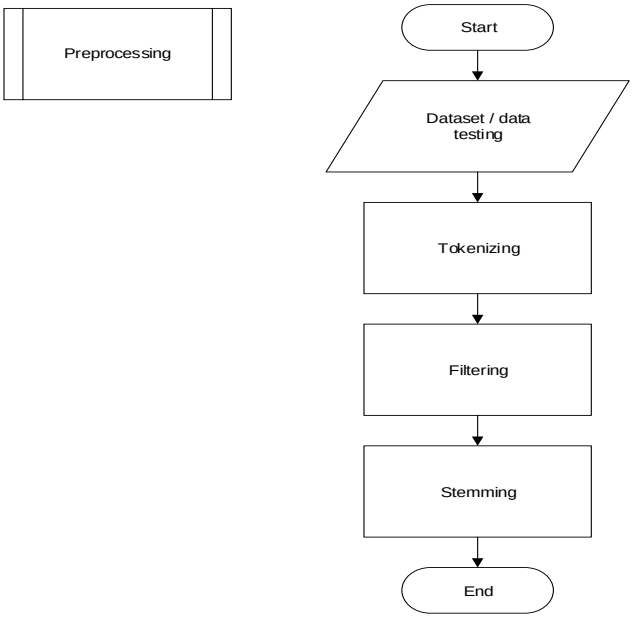
Gambar 4.3 Diagram Alir Sistem

Keterangan:

1. Mulai.
2. Disediakan data latih sejumlah 100 *tweets* berlabel serta data uji sebanyak 40 tidak berlabel.
3. Melakukan proses Preprocessing, yaitu *tokenizing, filtering*, dan *stemming*.
4. Melakukan proses pembobotan, yaitu mengalikan *Term Frequency* dengan *Inverse Document Frequency*.
5. Melakukan proses perhitungan kesamaan cosine antara data latih & data uji.
6. Melakukan tahapan klasifikasi *K-Nearest Neighbour.*
7. Selesai.

4.2.3.2 Diagram Alir Preprocessing

Berikut ini merupakan *flowchart preprocessing,* ditampilkan dalam Gambar 4.7.



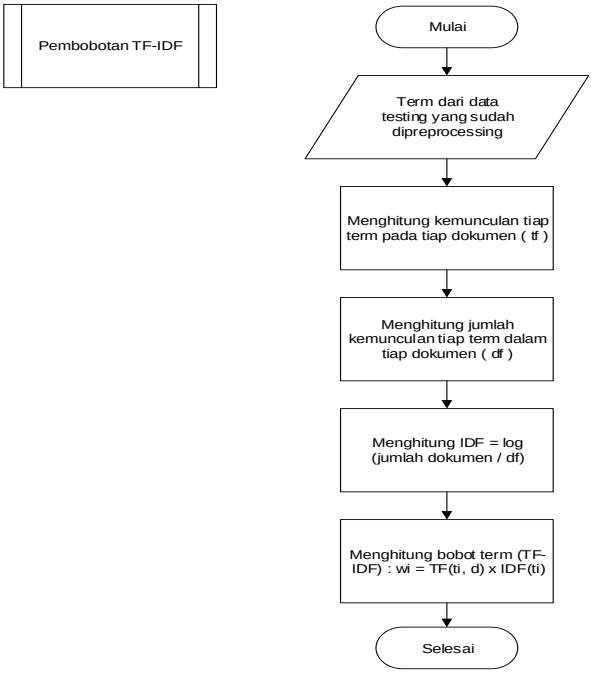
Gambar 4.4 Diagram Alir *Preprocessing*

Keterangan:

1. Mulai.
2. Disediakan dataset yang berasal dari *tweets* akun detik.com & kompas.com.
3. Melakukan proses *tokenizing*, yaitu upaya untuk memecah sekelompok karakter dalam suatu teks ke dalam satuan kata.
4. Melakukan proses *filtering*, yaitu dikerjakan penghilangan kosakata yang tidak bermanfaat, dikenal dengan istilah *stoplist* atau *stopword*, yaitu daftar kata yang kerap dipakai, tetapi bukan mendeskripsikan konten dokumen, seperti kata ”yang”, “dan”, “di”, “dari” dan seterusnya.
5. Melakukan proses *stemming,* menemukan dasar kata dari tiap kata dari hasil tahap sebelumnya.
6. Selesai.

4.2.3.3 Diagram Alir Pembobotan TF-IDF

Berikut ini merupakan *flowchart* dari kalkulasi TF.IDF, ditampilkan dalam Gambar 4.9.



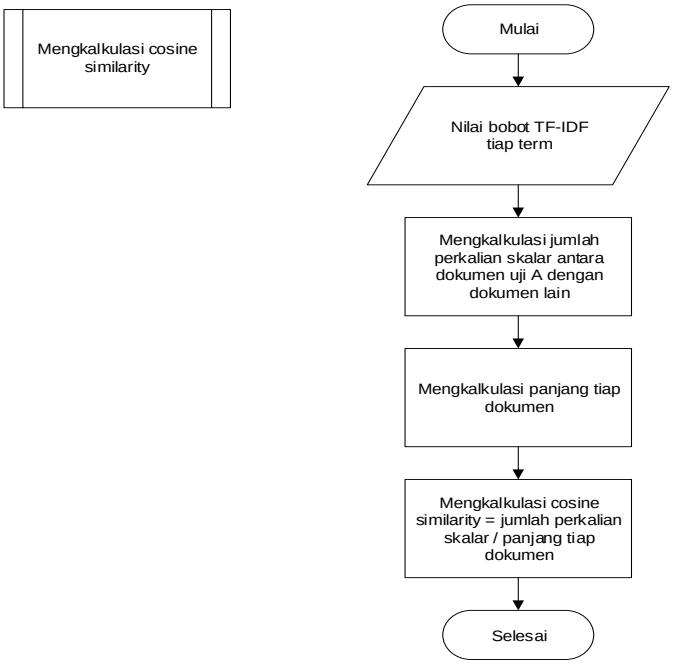
Gambar 4.5 Diagram Alir Pembobotan TF-IDF

Keterangan:

1. Mulai.
2. Disediakan *term* dari data uji yang sudah dilakukan proses preprocessing.
3. Melakukan proses perhitungan kemunculan *term* pada setiap dokumen.
4. Melakukan proses pengkalkulasian dokumen yang mengandung *term* tertentu.
5. Melakukan kalkulasi *Inverse Document Frequency* (IDF), untuk mengurangi dominasi kata yang kerap keluar dalam kumpulan dokumen.
6. Melaksanakan proses kalkulasi pembobotan *term*, dengan mengalikan frekuensi *term* dengan IDF yang telah dikerjakan sebelumnya.
7. Selesai.

4.2.3.4 Diagram Alir Perhitungan Cosine Similarity

Berikut ini merupakan *flowchart* kalkulasi kesamaan cosin, ditampilkan dalam Gambar 4.10.



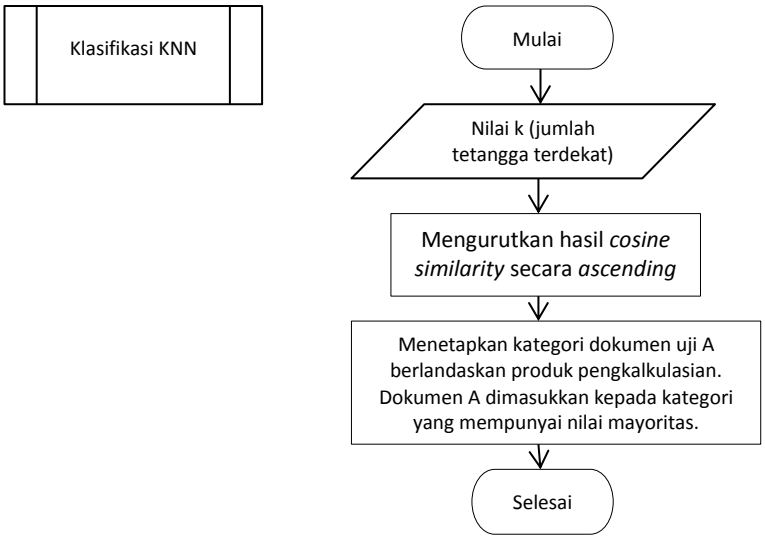
Gambar 4.6 Diagram Alir Kalkulasi *Cosine Similarity*

Keterangan:

1. Mulai.
2. Disiapkan data nilai bobot dari perkalian TF-IDF tiap *term.*
3. Melaksanakan perhitungan perkalian skalar, antara suatu dokumen data uji dengan dokumen latih lainnya.
4. Melakukan proses pengkalkulasian panjang tiap dokumen.
5. Melaksanakan proses perhitungan kesamaan cosine, yaitu dengan menjumlahkan perkalian panjang dokumen latih dan uji, lalu dibagi penjumlahan dari total bobot nilai dokumen uji & latih yang dikuadratkan kemudian diakar, selanjutnya dilakukan perkalian antara bobot tersebut.
6. Selesai.

4.2.3.5 Diagram Alir Klasifikasi KNN

Berikut ini merupakan *flowchart* pengklasifikasian *K-NN*, ditampilkan oleh Gambar 4.11.



Gambar 4.7 Diagram Alir Klasifikasi *K-NN*

Keterangan:

1. Mulai.
2. Disediakan nilai k yang akan diinputkan.
3. Melakukan proses pengurutan data hasil *cosine similarity* dari yang terbesar hingga terkecil.
4. Menentukan kategori data uji dari hasil perhitungan. Dokumen data uji dimasukkan ke kategori yang mempunyai nilai mayoritas.
5. Selesai.

4.3 Perancangan Antarmuka

Antarmuka (Interface) sebagai subsistem fungsinya menjadi alat interaksi antara pengguna dan sistem. Desain antarmuka penggunaan sistem ini sebagaimana diuraikan berikut ini.

4.3.1 Laman *Preprocessing* Teks

Di bagian awal ditunjukkan tabel, berisi dokumen fungsinya sebagai dataset. Sebelum menggunakan dataset, dilakukan lebih dulu proses *preprocessing* melalui tahap-tahap *case folding*, *tokenizing*, dan penghapusan *stopword* dan seluruhnya disajikan berupa kolom tabel.

Preprocessing | Term Unik | Klasifikasi

Browse doc.

id dok	Isi	Case Folding & Tokenizing	Stopword Removal
1	Jokowi sedang berkunjung ke luar negeri (2/12/2015)	jokowi sedang berkunjung ke luar negeri	jokowi kunjung luar negeri
2	Sepakbola Indonesia sudah lama vakum	sepak bola indonesia sudah lama vakum	sepak bola indonesia lama vakum
3	Pemerintah Jakarta merencanakan 3 in 1 untuk pengendara mobil	pemerintah jakarta merencanakan in untuk pengendara mobil	perintah jakarta rencana in kendara mobil
4			
5			
6			
7			

Gambar 4.8 Laman *Preprocessing* Teks

4.3.2 Laman *Term* Unik

Dalam dataset di laman kedua tampak susunan *term* unik. Munculnya kata unik, karena dilakukan tahapan *preprocessing* terlebih dahulu. Tabel di bawah ini menyajikan susunan id *term* dan kata unik.

Preprocessing | Term Unik | Klasifikasi

id term	term
1	jokowi
2	kunjung
3	luar
4	negeri
5	sepak
6	bola
7	indonesia
8	vakum
9	perintah
10	jakarta

Gambar 4.9 Laman *Term* Unik

4.3.3 Laman Klasifikasi

Pada halaman ini disajikan hasil klasifikasi *tweets* yang sudah dikerjakan. Selain *tweets* yang akan diklasifikasi, pada bagian ini juga muncul *account* berkaitan, dan keluaran pengklasifikasian kategori berdasarkan sistem.

Preprocessing | Term Unik | Klasifikasi

No.	Tweet	Akun	Kategori
1	Pemerintah mengunjungi luar negeri	@Pemerintah Indo	Politik
2	Tips menjaga ban motor awet	@IndoMotor	Otomotif
3	Selamat atas kemenangan Barcelona	@TIF_Sembiring	Olahraga
4	Jika hendak ke kota Malang jangan lupa berkunjung ke tempat ini	@IndoTravel	Travelling
5	Resep memasak sop ayam	@Rajamasak	Kuliner
6	Selamat atas kemenangan Real Madrid	@Bolanet	Olahraga
7	Peraturan 3 in 1 akan segera berlaku di Jakarta	@Pemerintah Indo	Politik

Gambar 4.10 Laman Klasifikasi

4.4 Contoh Perhitungan Manual

Pada sub bab ini ditunjukkan perhitungan manual dari program. Berikut contohnya:

4.4.1 Klasifikasi

1. Data training berupa *tweet* contohnya ada 4 *tweet* yaitu :

   Dok 1 : gejala infeksi virus zika | kategori: kesehatan

   Dok 2 : indonesia belum bebas tuberkolusis | kategori: kesehatan

   Dok 3 : menyusutnya otak partisipan | kategori: kesehatan

   Dok 4 : prestasi bulutangkis indonesia menurun |kategori: olahraga

2. Data testing misalnya :

   Dok x : infeksi virus menurun

3. Selanjutnya semua data di*preprocessing* dahulu:

   Data training :

   Dok 1 : gejala infeksi virus zika | kategori: kesehatan

   Dok 2 : indonesia bebas tuberkolusis | kategori: kesehatan

Dok 3 : susut otak partisipan | kategori: kesehatan

Dok 4 : prestasi bulutangkis indonesia turun | kategori: olahraga

Data testing :

Dok x : infeksi virus turun

4. Langkah selanjutnya yaitu menghitung bobot TF-IDF dari tiap-tiap *term*

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc X	df	idf	Tf-idf				
								1	2	3	4	X
infeksi	1	0	0	0	1	2	0,30103	0,30103	0	0	0	0,30103
virus	1	0	0	0	1	2	0,30103	0,30103	0	0	0	0,30103
zika	1	0	0	0	0	1	0,60206	0,60206	0	0	0	0
indonesia	0	1	0	1	1	3	0,124939	0	0,124939	0	0,124939	0,124939
bebas	0	1	0	0	0	1	0,60206	0	0,60206	0	0	0
tuberkolusis	0	1	0	0	0	1	0,60206	0	0,60206	0	0	0
susut	0	0	1	0	0	1	0,60206	0	0	0,60206	0	0
otak	0	0	1	0	0	1	0,60206	0	0	0,60206	0	0
bulutangkis	0	0	0	1	0	1	0,60206	0	0	0	0,60206	0
turun	0	0	0	1	1	2	0,30103	0	0	0	0,30103	0,30103

5. Setelah itu proses berikutnya menghitung *cosine similarity*

Sebelumnya, hitung perkalian term dokumen uji dengan dokumen latih. Lalu hitung panjang dokumen, dengan mencari akar dari jumlah kuadrat tiap term dokumen.

Perkalian skalar dokumen x dengan tiap dokumen				
Term	**Dokumen**			
	1	**2**	**3**	**4**
infeksi	0,09062	0	0	0
virus	0,09062	0	0	0
zika	0	0	0	0
indonesia	0	0,01561	0	0,01561
bebas	0	0	0	0
tuberkolusis	0	0	0	0
susut	0	0	0	0
otak	0	0	0	0
bulutangkis	0	0	0	0
turun	0	0	0	0,09062
SUM	0,18124	0,01561	0	0,10623

	Panjang Dokumen				
	1	2	3	4	x
SUM	0,01642	0,00024	0	0,00845	0,28747
SQRT	0,1281	0,0154	0	0,0919	0,5361

Kesamaan cosin (Cosine Similarity)

Dokumen	Indeks kelas	Cosine Similarity
1	1	2,639119870
2	1	1,890759865
3	1	0
4	2	2,156184217

6. Kemudian menetapkan *k values* untuk tetangga yang dekat, misalnya k = 3 lalu diurutkan berdasarkan jumlah *Cosine Similarity* terbesar sampai yang terkecil.

Dokumen	Indeks kelas	Cosine Similarity
1	1	2,639119870
4	2	2,156184217
2	1	1,890759865

Karena jumlah indeks kelas 1 lebih banyak, maka dokumen x menjadi anggota kategori 1, oleh karena itu *tweet* itu masuk pada kategori kesehatan.

4.5 Perancangan Uji Coba

Sub bab ini akan dirancang mencari nilai akurasi yang terbaik. Selanjutnya dibandingkan guna menguji akurasi dengan metode sebagaimana disajikan pada tabel berikut ini.

Klasifikasi	**Hasil Akurasi**
K-NN dengan nilai *k* <= 3	
K-NN dengan nilai *k* >= 5	

Tabel 4.6 Perencanaan Metode Percobaan

BAB 5 IMPLEMENTASI

Sebagai tindak lanjut dari uraian dan penjelasan sebelumnya tentang metodologi dan perancangan, maka dalam Bab 5 ini masuk pada bagian implementasi yang substansinya mencakup: pertama, batasan implementasi itu sendiri; kedua, tentang implementasi algoritme; dan ketiga, tentang implementasi *interface*.

5.1 Batasan Implementasi

Di bagian ini dideskripsikan mengenai batasan terkait dengan cara bekerjanya sebuah sistem yang mengacu pada perancangan yang sudah dibahas di bagian terdahulu. Tujuan dari batasan ini agar sistem yang dikembangkan sejalan dan tidak menyimpang dari tujuan utamanya. Batasan yang dimaksud meliputi:

1. Penggunaan Metode K-Nearest Neighbour dalam klasifikasi *Tweets* pada Twitter, dirancang & dijalankan dengan aplikasi berbahasa JAVA.
2. Metodologi pengerjaan permasalahan yang dipakai yakni K-Nearest Neighbour.
3. Data yang dipergunakan selaku data *training* & data *testing* adalah *tweets* yang diambil sumbernya dalam portal kompas & detik.
4. *Output* yang ditunjukkan adalah produk pengklasifikasian yang ada dalam kelompok teknologi, ekonomi, kesehatan, olahraga, serta otomotif.

5.2 Implementasi Algoritme

Implementasinya dibagi menjadi 3 algoritme yakni, *Text Preprocessing,* pembobotan dan klasifikasi. Penjelasannya sebagai berikut.

5.2.1 *Text Preprocessing*

Algoritme *text preprocessing* adalah langkah pertama yang dikerjakan sistem, kemudian dilanjutkan dengan tahapan klasifikasi. Tahap-tahapnya yakni proses *case folding, tokenizing, filtering,* terakhir proses *stemming*.

5.2.1.1 *Case folding & Tokenizing*

Case Folding & *Tokenizing* adalah teknik yang dilakukan untuk mengganti karakter abjad ukuran besar atau yang mengandung huruf kapital (*upper case*) jadi abjad biasa (*lower case*) serta menguraikan tiap perkataan menjadi banyak penggalan kata.

```
1  public String casekecilkan(String masukkanFile) throws
2         FileNotFoundException, IOException{
3            return masukkanFile.toLowerCase();
```

```
4          }
5   public   static   String   menghapusAngkadanTandaBaca(String
6  kalimat) {
7           kalimat = kalimat.replaceAll("\\p{Punct}|\\d", " ");
8           return kalimat;
9          }
10
11  public ArrayList<String> token (String kalimat) {
12      String produkKalimat =
13      menghapusAngkadanTandaBaca(kalimat);
14      String hapuskanNewline =
15      produkKalimat.replace(System.getProperty("line.separator
16      "), " ");
17      daftarKata = new ArrayList<String>();
18         token = new StringTokenizer(hapuskanNewline, " ");
19         while (token.hasMoreTokens()) {
20               words = token.nextToken();
21               daftarKata.add(words);
22         }
23         return daftarKata;
24     }
```

Kode Program 5.1 Screenshot kode *case folding* & *tokenizing*

Keterangan:

Baris 1 - 4 : Deklarasi method & penangkapan kesalahan.
5 - 9 : Deklarasi method hapus angka & tanda baca, diganti dengan spasi.
11 : Membuat Arraylist tipe String dengan memasukkan parameter "kalimat".
12 - 16 : Menginisiasi atribut bertipe String yang berfungsi menghapus angka & tanda baca. Mengganti *line* baru dengan spasi.
17 : Menginisialiasi atribut & dimasukkan ke Arraylist bertipe String.
18 : Menginisialisasi variabel yang berfungsi menghapus *line* baru.
19 - 24 : Melakukan iterasi untuk dimasukkan ke daftar kata & mengembalikan nilai ke variabel daftarKata.

5.2.1.2 *Filtering*

Langkah ini merupakan operasi yang melakukan penghilangan istilah tak bermanfaat (*stoplist*) atau bisa dikatakan menyimpan istilah berguna (*wordlist*).

```
1    public String pembersihanStopword(String inputdok) throws
2              FileNotFoundException, IOException {
3       String liststopword = bacaDokumenText("Stopwords.txt");
4         String berhenti = "\\b(" + liststopword + ")\\b\\s*";
5                 Pattern stopwords = Pattern.compile(stop,
6                    Pattern.CASE_INSENSITIVE);//
7              Matcher cocok = stopwords.matcher(inputdok);
8                     pembersih = cocok.replaceAll("");
9                         return pembersih;
10      }
```

Kode Program 5.2 Screenshot kode *filtering*

Keterangan:

Baris 1 - 2 : Deklarasi method menghapus Stopword dengan melewatkan parameter bertipe String & penangkapan kesalahan
3 : Membuat variabel yang berfungsi membaca dokumen bertipe txt.
4 : Membuat variabel bertipe String untuk menemukan stopword.
5 - 6 : Membuat variabel stopwords & mengatur stopwords agar *case*nya *insensitive.*
7 : Mencocokkan stopwords dengan parameter method.
8 - 9 : Membuat variabel yang berfungsi mengganti stopword dengan string kosong.

5.2.1.3 *Stemming*

Langkah ini berupa prosedur menemukan dasar kata dari tiap kata hasil *filtering*.

```
1  public ArrayList<String> stemming( ArrayList<String> kalimat)
2          throws IOException{
3    StemmingNaziefAndriani stem = new StemmingNaziefAndriani();
4           for (int p = 0; p < listKata.size(); p++) {
5              listKata.set(p, stem.KataDasar(listKata.get(p)));
6           }
7           return listKata;
8      }
9
```

Kode Program 5.3 Screenshot kode *stemming*

Keterangan:

Baris 1 - 2 : Deklarasi method Arraylist bertipe String dan melewatkan argumen, ditambahi dengan penangkapan kesalahan.
3 : Menginisialisasi variabel dengan mereferensi kelas NaziefAndriani.
4 - 8 : Melakukan iterasi dimulai dari nol hingga ukuran listKata, kemudian dicari kata dasarnya. Mengembalikan variabel listKata.

5.2.2 Pembobotan TF.IDF

Kelaziman untuk menampilkan sebuah dokumen yang tersaji dalam *vector space model*, biasanya menggunakan metode Pembobotan TF.IDF, terutama jika dikaitkan dengan pembelajaran yang menggunakan metode K-NN dan SVM, apalagi dalam konteks klasifikasi teks.

```
1      double akarIdf = 0.0;
2      double TFlatih = 0.0;
3      double TFIDFlatih = 0.0;
4             for (int m = 0; m < dLat.length; m++) {
5                if (1.0 + (Math.log10(mtfLatih.getDt(i, m))) ==
6                    Double.NEGATIVE_INFINITY) {
7                TFlatih = 0.0;
8             TFIDFlatih = TFlatih * (dataWeight[0]);
```

```
9                } else {
10             TFlatih = Math.log10(mtfLatih.getDt(i, m));
11                    TFlatih = 1.0 + TFlat;
12                   TFIDFlatih = TFlatih * (dataWeight[0]);
13               }
14               logFrek_lat[m] = String.valueOf((TFlat));
15             Tf_IDf[m] = String.valueOf((TFIDFlatih));
16              wIdfLat[i][m] = Double.valueOf(tf_idf[m]);
17                   akarIdf += wIdfLat[i][m];}
18
```

Kode Program 5.4 Screenshot kode *stemming*

Keterangan:

Baris 1 - 3 : Membuat atribut bertipe *double* dan diset 0.0.
4 : Melakukan iterasi dari indeks 0 hingga mencapai panjang data latih.
5 - 6 : Melakukan fungsi percabangan IDF.
7 - 8 : Frekuensi term latih bernilai nol, term data latih dikalikan bobot data.
9 - 13 : Term data latih didapatkan dengan menghitung log lalu ditambah satu. TF-IDF data latih didapatkan dengan mengalikan term dengan bobotnya.
14 : Mencari frekuensi term data latih.
15 - 17 : Membuat variabel TF-IDF yang mencari nilai dari String data tersebut. Mencari bobot IDF latih dari tf_idf. Akar IDF untuk menjumlahkan bobot IDF latih.

5.2.3 Klasifikasi K-Nearest Neighbour

Klasifikasi berdasarkan metode K-Nearest Neighbour, implementasinya seperti tertera di bawah ini.

```
1   System.out.println("COSINE SIMILARITY DOK UJI DAN LATIH");
2                 double cosTetaIdf[][] = new
3              double[dLatih.length][dUji.length];
4         String cosIdf[] = new String[dLatih.length];
5                   Integer CtgrIdf[][] = new
6              Integer[dLatih.length][dUji.length];
7                   Integer indeksIdf[][] = new
8              Integer[dLatih.length][dUji.length];
9
10           for (int o = 0; o < dLat.length; o++) {
11                      int colu_uji = 0;
12             for (int n = 0; n < dU.length; n++) {
13           CtgrIdf[o][n] = dataList_katlatih.get(i);
14                  indeksIdf[o][n] = (o + 1);
15    cosTetaIdf[o][n] = vecIdfQ_D[o][n] / (akarIdfUji[colu_uji]
16                      * akarIdfDokLAt[o]);
17             if (Double.isNaN(cosTetaIdf[o][n])) {
18                      cosTetaIdf[o][n] = 0.0;
19                              }
20         cosIdf[n] = String.valueOf(cosTetaIdf[o][n]);
21            System.out.print(CtgrIdf[o][n] + "|");
22           System.out.print(indeksIdf[o][n] + "= ");
```

```
23          System.out.print(((cosTetaIdf[o][n])) + "\t");
24                              colu_uji++;
25                  }
26                          IdfCosin.addRow(cosIdf);
27                          System.out.println("");
28          }
29
30          DefaultTableModel sortIdfSim = (DefaultTableModel)
31                          sortIdf.getModel();
32                  for (int q = 0; q < dU.length; q++) {
33                     sortIdfSim.addColumn("Query " + (q + 1));
34                  }
35
36          String sort_cosIdf[] = new String[dLat.length];
37                          System.out.println("");
38            System.out.println("SORTING NILAI TERBESAR");
39                      for (int r = 0; r < dLatih.length; r++) {
40                          for (int q = 0; q < dUji.length; q++) {
41          sorting_cosIdf[i] = String.valueOf(cosTetaIdf[r][q]);
42                  System.out.print(CtgrIdf[r][q] + "|");
43                  System.out.print((indeksIdf[r][q]) + "= ");
44                  System.out.print(((cosTetaIdf[r][q])) + "\t");
45                      }
46                      sortIdfSim.addRow(sorting_cosIdf);
47                      System.out.println();
48                  }
49
50          DefaultTableModel Idf_k = (DefaultTableModel)
51                  k_Idf.getModel();
52                  for (int i = 0; i < dU.length; i++) {
53                      Idf_k.addColumn("Query " + (i + 1));
54                  }
55
56          String K_SimIdf[] = new String[dLat.length];
57                  //===
58                  int K = Integer.parseInt(jTextField1.getText());
59                  System.out.println("");
60            System.out.println("NILAI K = "+Integer.toString(K));
61            for (int j = 0; j < K; j++) {
62                  for (int i = 0; i < dU.length; i++) {
63             System.out.print(CtgrIdf[j][i] + "|");
64             System.out.print((indekIdf[j][i]) + "= "); //Penting
65             K_SimIdf[i] = String.valueOf(cosTetaIdf[j][i]);
66             System.out.print(cosTetaIdf[j][i] + "\t");
67                  }
68                      Idf_k.addRow(K_SimIdf);
69                      System.out.println();
70          }
```

Kode Program 5.5 Screenshot kode K-NN

Keterangan:

Baris 1 : Membuat method cetak string

2 -3 : Inisialisasi atribut cosine similarity

4 : Inisialisasi String data latih

5 - 8 : Inisialisasi kategori & indeks yang berisi array dari data latih & uji.

10 - 28 : Dilakukan input data latih & data uji. Kemudian dicari cosine similarity-nya. Dilakukan dengan menambah baris dalam setiap input data latih & ujinya.
30 - 34 : Membuat tabel untuk mengurutkan kesamaan cosine, mulai dari yang terbesar hingga terkecil.
36 - 48 : Membuat method untuk mengurutkan kesamaan cosine, dari data latih serta data uji. Dilakukan dengan menambah baris dalam tiap inputnya.
50 - 54 : Membuat tabel untuk nilai k. Dan tiap inputnya akan ditambah baris.
56 - 70 : Membuat method untuk nilai k yang bisa diinputkan melalui antarmuka (*Interface*). Dilakukan dengan menambah baris tiap inputnya.

5.2.4 Menghitung Akurasi

Proses ini menunjukkan perhitungan seberapa akurat dokumen terklarifikasi dengan benar.

```
1  double acuracyIDF = (100.0 / dU.length) * IdfTrue;
2        System.out.println("Akurasi KNN = " + acuracyIDF + "%");
3
```

Kode Program 5.6 Screenshot kode akurasi

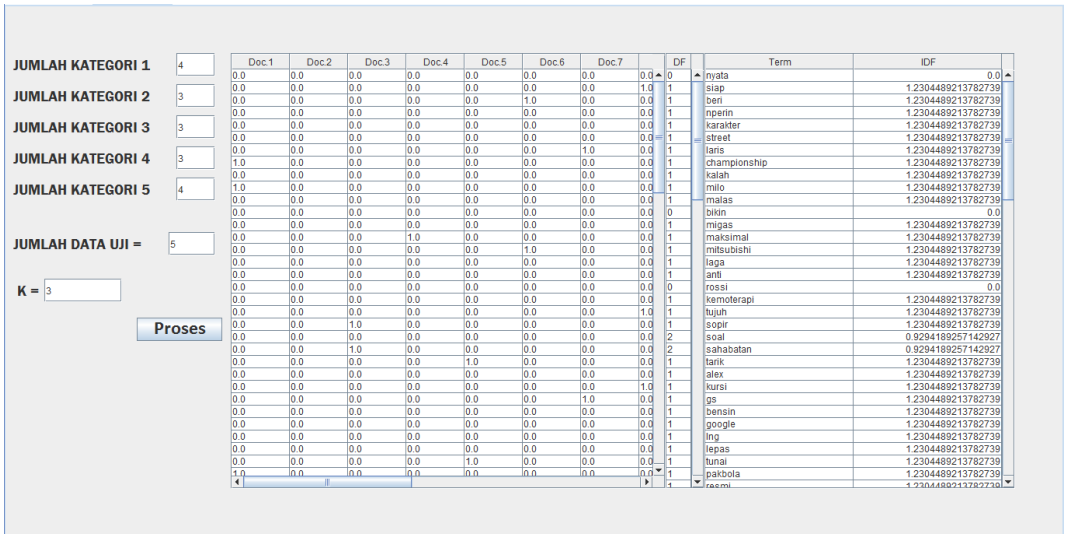
Keterangan:

Baris 1 : Membuat variabel akurasi & membuat persamaannya.
2 : Melakukan method cetak akurasi ditambah lambang %.

5.3 Implementasi Antarmuka

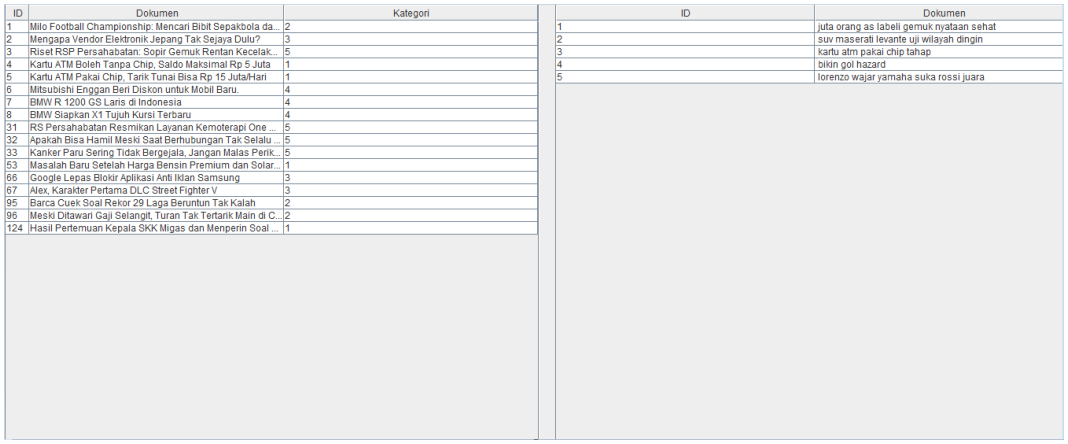
Antarmuka (*Interface*) program bertujuan agar *user* dan sistem dapat berinteraksi langsung. *Interface* yang dapat ditampilkan antara lain *preprocessing,* penghitungan kemunculan jumlah *term*, pembobotan, dan proses klasifikasi.

Laman ini menampilkan perhitungan proses Tf-Idf dan input besaran data *training*, data *testing*, beserta *k values*. Implementasinya yaitu ditunjukkan oleh Gambar 5.1.



Gambar 5.1 Screenshot tampilan Input

Data latih serta data uji yang digunakan pada pengujian. Implementasinya yakni ditampilkan dalam Gambar 5.2.



ID	Dokumen	Kategori
1	Milo Football Championship: Mencari Bibit Sepakbola da...	2
2	Mengapa Vendor Elektronik Jepang Tak Sejaya Dulu?	3
3	Riset RSP Persahabatan: Sopir Gemuk Rentan Kecelak...	5
4	Kartu ATM Boleh Tanpa Chip, Saldo Maksimal Rp 5 Juta	1
5	Kartu ATM Pakai Chip, Tarik Tunai Bisa Rp 15 Juta/Hari	1
6	Mitsubishi Enggan Beri Diskon untuk Mobil Baru.	4
7	BMW R 1200 GS Laris di Indonesia	4
8	BMW Siapkan X1 Tujuh Kursi Terbaru	4
31	RS Persahabatan Resmikan Layanan Kemoterapi One ...	5
32	Apakah Bisa Hamil Meski Saat Berhubungan Tak Selalu ...	5
33	Kanker Paru Sering Tidak Bergejala, Jangan Malas Perik...	5
53	Masalah Baru Setelah Harga Bensin Premium dan Solar...	1
66	Google Lepas Blokir Aplikasi Anti Iklan Samsung	3
67	Alex, Karakter Pertama DLC Street Fighter V	3
95	Barca Cuek Soal Rekor 29 Laga Beruntun Tak Kalah	2
96	Meski Ditawari Gaji Selangit, Turan Tak Tertarik Main di C...	2
124	Hasil Pertemuan Kepala SKK Migas dan Menperin Soal ...	1

ID	Dokumen
1	juta orang as labeli gemuk nyataan sehat
2	suv maserati levante uji wilayah dingin
3	kartu atm pakai chip tahap
4	bikin gol hazard
5	lorenzo wajar yamaha suka rossi juara

Gambar 5.2 Screenshot tampilan Data latih & uji

Laman ini menyajikan model ruang vektor dihasilkanlah *preprocessing* & kalkulasi Frekuensi Term (atau dilambangkan TF_i), Frekuensi Dokumen (dilambangan dengan DF_i), Frekuensi Dokumen Invers (bisa disingkat menjadi IDF) dimana dipakai di tahap TF-IDF *weighting*. Implementasinya ditampilkan oleh Gambar 5.3.

Doc.1	Doc.2	Doc.3	Doc.4	Doc.5	Doc.6	Doc.7	Doc.8	Doc.9	Doc.10	Doc.11	Doc.12	Query 1	Query 2	Query 3	Query 4	Query 5
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

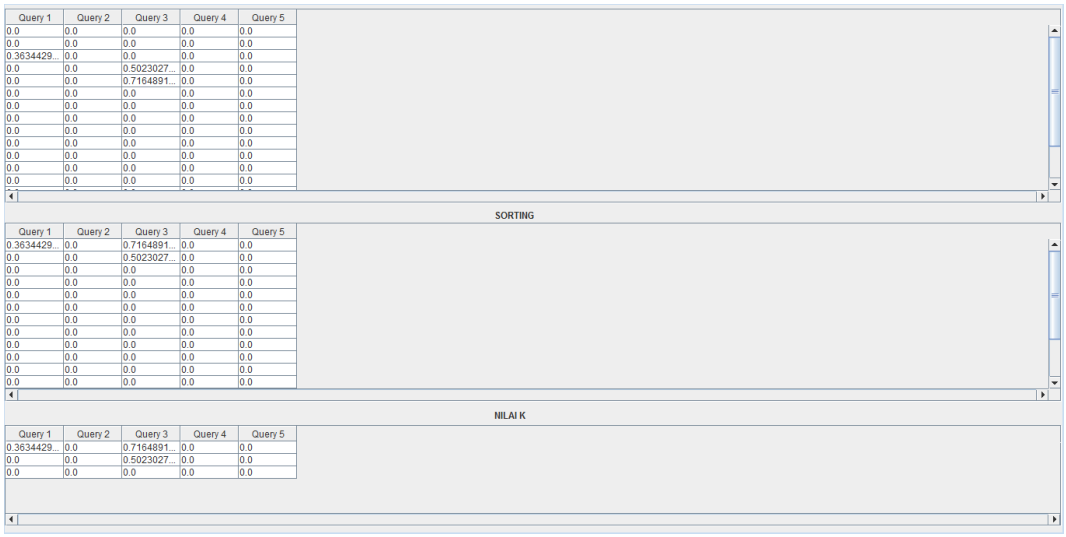
Gambar 5.3 Screenshot tampilan frekuensi term

Laman ini menunjukkan model ruang vektor produk kalkulasi TF-IDF *weighting*. Implementasinya ditampilkan dalam Gambar 5.4.

Doc.1	Doc.2	Doc.3	Doc.4	Doc.5	Doc.6	Doc.7	Doc.8	Doc.9	Doc.10	Doc.11	Doc.12	Query 1	Query 2	Query 3	Query 4	Query 5
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.9294189...	0.0	0.0	0.0	0.0	0.0	0.9294189...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.2304489...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3.2554618...	2.1312000...	3.3855356...	2.5462300...	2.8279483...	2.4608978...	2.3250447...	2.9041078...	3.3855356...	2.1312000...	2.9357694...	2.4608978...	1.2304489...	0.0	2.0261942...	0.0	0.0

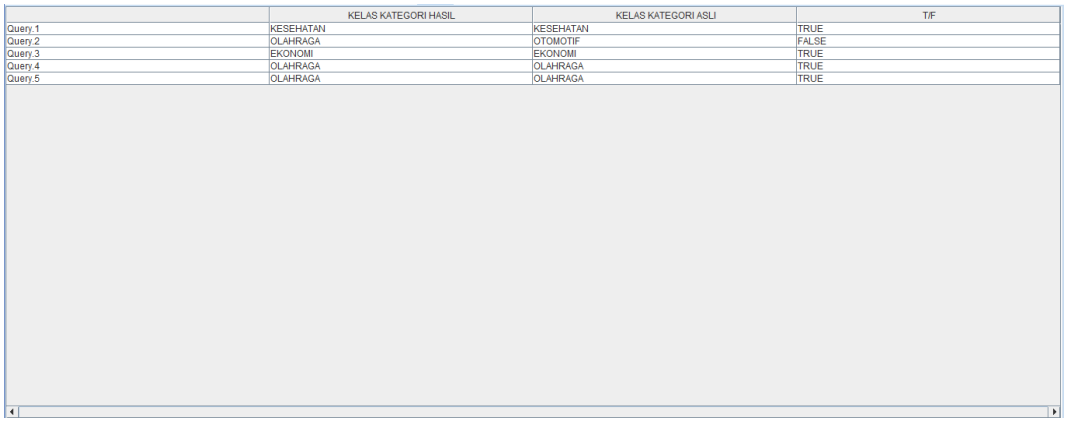
Gambar 5.4 Screenshot tampilan perhitungan TF-IDF

Berikutnya adalah tampilan model ruang vektor sebagai konsekuensi dari hitungan *cosine similarity* yang dihitung yang mengacu pada data yang diuji beserta data yang dilatih, berasal daripada TF-IDF *weighting* yang selanjutnya menetapkan besaran dari *k*, dimana Implementasinya seperti ditampilkan di Gambar 5.5.



Gambar 5.5 Screenshot tampilan *Cosine Similarity*

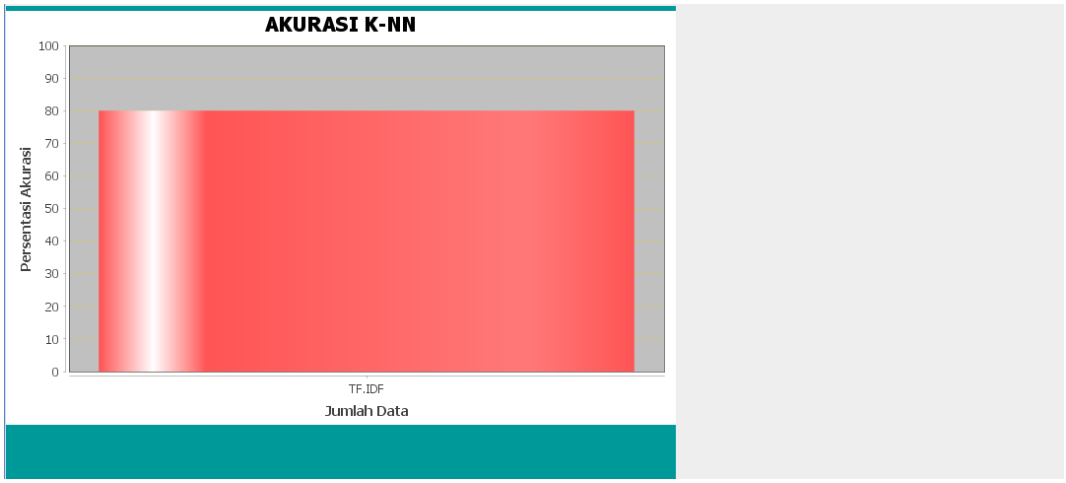
Laman di bawah ini menampilkan hasil klasifikasi. Implementasinya ditunjukkan pada 5.6.



	KELAS KATEGORI HASIL	KELAS KATEGORI ASLI	T/F
Query.1	KESEHATAN	KESEHATAN	TRUE
Query.2	OLAHRAGA	OTOMOTIF	FALSE
Query.3	EKONOMI	EKONOMI	TRUE
Query.4	OLAHRAGA	OLAHRAGA	TRUE
Query.5	OLAHRAGA	OLAHRAGA	TRUE

Gambar 5.6 Screenshot tampilan hasil klasifikasi

Tingkat keakuratan dari kalkulasi *K-Nearest Neighbour* yang dihasilkan, implementasinya seperti terlihat di gambar 5.7.



Gambar 5.7 Screenshot tampilan akurasi

BAB 6 PENGUJIAN & ANALISIS

Pemakaian kaidah K-Nearest Neighbour (K-NN) untuk menguji serta menganalisis klasifikasi *Tweets* pada Twitter disajikan pada pembahasan berikut ini.

6.1 Pengujian Metode K-Nearest Neighbour (K-NN)

Pengujian atas penggunaan kaidah K-*Nearest Neighbour* (K-NN) mencakup 2 (dua) hal, yakni pertama, berkenaan dengan skenario pengujian dan kedua, tentang analisa pengujian.

6.1.1 Skenario Pengujian

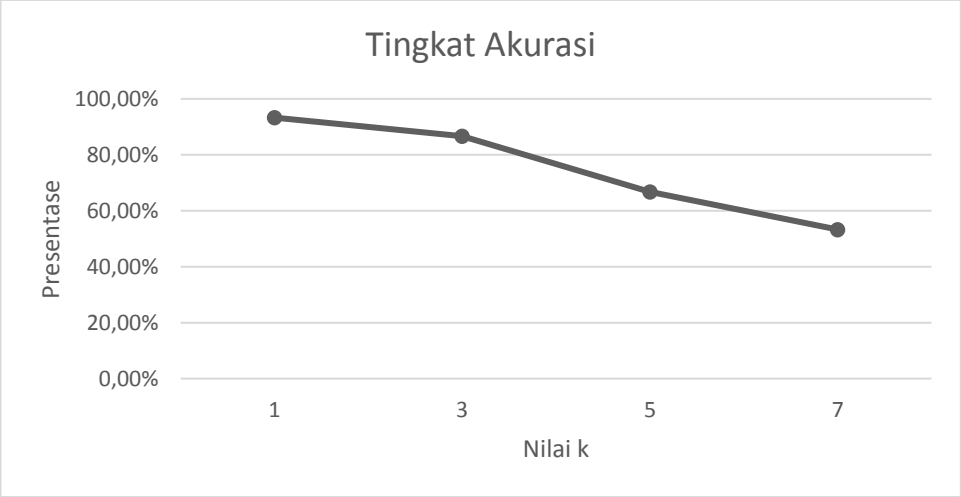
Untuk mengetahui keakuratan dari metode ini, maka dilakukan skenario pengujian dengan mengambil total data ada 65, dengan rincian data uji yang dipakai sejumlah 15 data, dengan data *random*. Sedangkan untuk data latih yang dipakai sejumlah 10 data dari setiap kategori, atau 50 data buat seluruh kategori. Lalu *k values* dimasukkan yaitu 1, 3, 5, serta 7.

K = 1, Presentase 93,3%, Dokumen benar terklasifikasi ada 14

K = 3, Presentase 86,7%, Dokumen benar terklasifikasi ada 13

K = 5, Presentase 66,7%, Dokumen benar terklasifikasi ada 10

K = 7, Presentase 53,3%, Dokumen benar terklasifikasi ada 8



Gambar 6.1 Grafik akurasi dengan 50 data latih

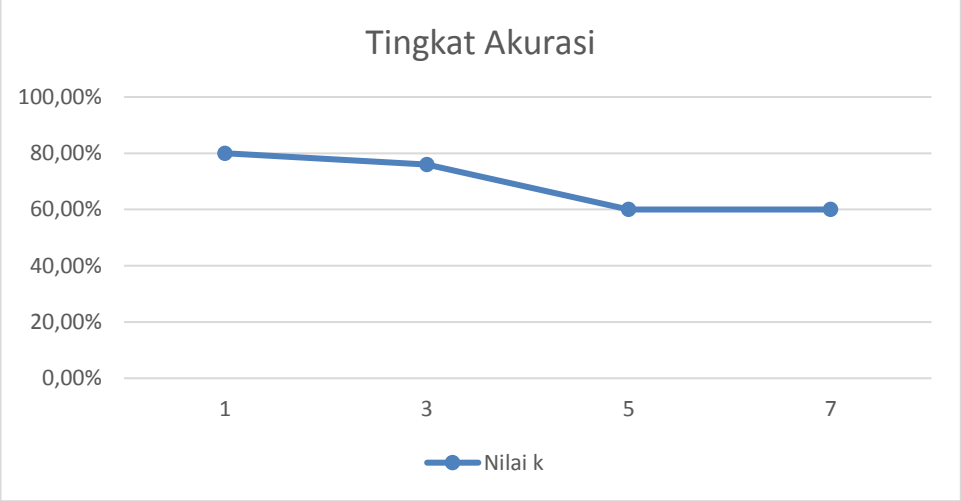
Pengujian selanjutnya dengan jumlah data yang berbeda, yaitu total datanya 100, dengan uraian 75 data latih, 15 data untuk tiap kategori, dan 25 data uji. Selanjutnya nilai *k* yang dimasukkan yakni 1, 3, 5, dan 7.

K = 1, Presentase 80,0 %, Dokumen benar terklasifikasi ada 20

K = 3, Presentase 76,0%, Dokumen benar terklasifikasi ada 19

K = 5, Presentase 60,0%, Dokumen benar terklasifikasi ada 15

K = 7, Presentase 60,0%, Dokumen benar terklasifikasi ada 15



Gambar 6.2 Grafik akurasi dengan 75 data latih

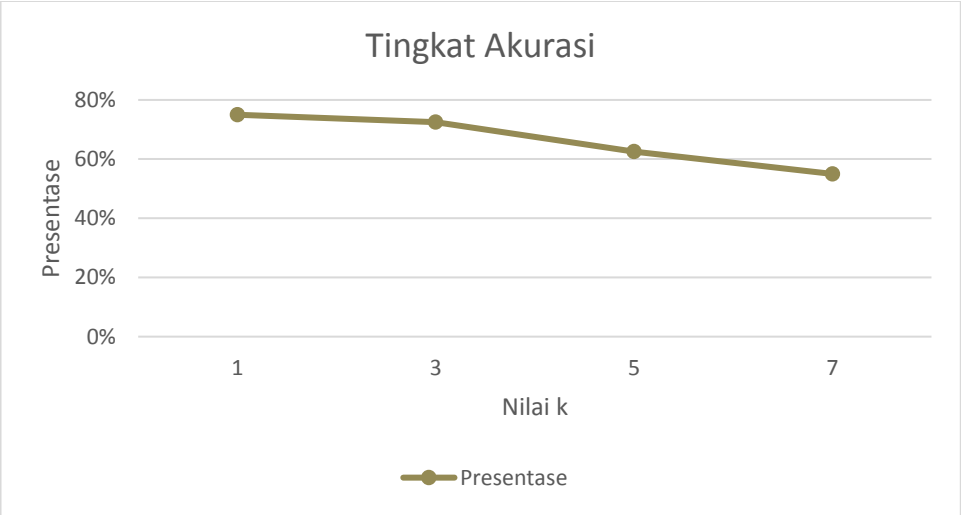
Data yang dipakai pada pengujian berikutnya jumlahnya lebih banyak, yakni total data 140, dengan uraian 100 data latih serta 40 data uji. Berikutnya nilai *k* yang dimasukkan yaitu 1, 3, 5, serta 7.

K = 1, Presentase 75,0%, Dokumen benar terklasifikasi ada 30

K = 3, Presentase 72,5%, Dokumen benar terklasifikasi ada 29

K = 5, Presentase 62,5%, Dokumen benar terklasifikasi ada 25

K = 7, Presentase 55,0%, Dokumen benar terklasifikasi ada 22



Gambar 6.3 Grafik akurasi dengan 100 data latih

6.1.2 Analisis Pengujian

Keakuratan hasil uji penggunaan kaidah *K-Nearest Neighbour* (K-NN), menunjukkan bahwasannya keakuratan menurun dibarengi semakin besarnya nilai *k* dan bertambahnya jumlah data. Seperti percobaan pertama dengan total data sebanyak 65, dimana jumlah data latih sebesar 50 data serta data uji sebesar 15 data. Dari pengujian tersebut menunjukkan tingkat akurasi ketika nilai *k* = 1 atau 3 lebih besar dibandingkan dengan nilai *k* lainnya. Hal ini membuktikan bahwasannya nilai *k* mempunyai efek terhadap prosedur klasifikasi memakai metode *K-Nearest Neighbour* (K-NN), lantaran bisa disebabkan oleh penyebaran data yang berbeda dan dalam hal tertentu kelas asli mendominasi nilai k atau bahkan kebalikannya.

BAB 7 PENUTUP

Berlandaskan pembahasan bab-bab sebelumnya, dapat diambil konklusi serta beberapa masukan yang diharapkan sanggup digunakan bagi penelitian kedepannya.

7.1 Kesimpulan

Kesimpulan atas penggunaan metode *K-Nearest Neighbour* (K-NN) dengan TF-IDF *weighting* untuk menguji dan menganalisis klasifikasi *tweets* pada Twitter dapat disajikan berikut ini.

1. Penerapan metode K-NN pada *tweets* pengguna Twitter yakni dengan memasukkan *tweets* tersebut pada dokumen uji, lalu selanjutnya dibandingkan dengan data yang ada pada dokumen latih. Kemudian akan diketahui hasil klasifikasi dari *tweets* tersebut.
2. Cara memaksimalkan klasifikasi memakai metode K-NN yaitu menginputkan *k values* dengan nilai yang tidak terlalu besar, karena bila nilai *k* semakin besar, maka jumlah dokumen yang terklasifikasi benar akan berkurang.
3. Semakin kecil penggunaan nilai k, maka semakin akurat penggunaan metode K-NN. Begitu pula sebaliknya, jika penggunaan nilai *k* bertambah besar, maka tingkat akurasi yang dihasilkan akan cenderung menurun.

7.2 Saran

Saran yang dapat diberikan guna pengembangan observasi mendatang, diantaranya adalah:

1. Diperlukan pengujian klasifikasi menggunakan algoritme klasifikasi lain atau modifikasi, seperti kaidah *Fuzzy C-Means*, *Improved Naïve Bayes,* atau metode lain guna mengetahui metode mana yang mempunyai akurasi lebih baik.
2. Direkomendasikan perlunya penelitian lanjutan dalam rangka menghasilkan klasifikasi yang semakin presisi dengan penggunaan algoritme stemming yang semakin sempurna, karena *text preprocessing* yang diterapkan dalam penelitian ini dirasakan masih belum optimal, misalnya proses stemming belum cukup efektif menghilangkan awalan & akhiran pada dokumen.

DAFTAR PUSTAKA

Adeniyi, D., Wei, Y., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using. Applied Computing and Informatics, 12, 90–108.

Asian, J., (2007). Effective Techniques for Indonesian Text Retrieval. S3. School of Computer Science and Information Technology, Science, Engineering, and Technology Portfolio, RMIT University, Melbourne, Victoria, Australia.

Baoli, Li., Shiwen, Yu., dan Qin, Lu. (2003). An Improved k-Nearest Neighbors for Text Categorization. In: Computer Network and Distributed Systems Laboratory. *Proceedings of the 20th International Conference of Computer Processing of Oriental Language.* Shenyang, China, 2003. Peking University: Department of Computer Science and Technology.

Fachruddin, M. (2011). Analisis dan Implementasi Pseudo Relevance Feedback dengan Kueri Expansion Menggunakan Term Selection Value. [e-journal].

Garcia, E., Dr. (2005). Latent Semantic Indexing (LSI) A Fast Track Tutorial. [e-journal].

Hardiyanto, E., & Rahutomo, F. (2016). Studi Awal Klasifikasi Artikel Wikipedia Bahasa Indonesia Dengan Menggunakan Metoda K-Nearest Neighbor. Seminar Nasional Terapan Riset Inovatif Semarang. Semarang.

J. Dixion, Dr. Brian. (2012). *Social Media for School Leader*. [e-book] John Wiley & Sons.

Mandala, R. dan Hendra, S., (2002). Improving Information Retrieval System Performance by Automatic Query Expansion. Universitas Padjadjaran Bandung.

Nurjanah, W. Perdana, S. Dan Fauzi, M. (2017). Analisis Sentimen Terhadap Tayangan televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode *K-Nearest Neighbor* dan Pembobotan Jumlah *Retweet.* Fakultas Ilmu Komputer. Universitas Brawijaya Malang.

Perdana, R.S., (2013). Pengkategorian Pesan Singkat Berbahasa Indonesia Pada Jejaring Sosial Twitter Dengan Metode Klasifikasi Naïve Bayes. S1. *Program Teknologi Informasi dan Ilmu Komputer,* Universitas Brawijaya.

Phuvipadawat, S. and Murata, T. (2010). Breaking News Detection and Tracking in Twitter. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. 31 Agustus – 3 September 2010. Washington DC: IEEE Computer Society.

Putri, P.A., (2013). Implementasi Metode Improved K-Nearest Neighbor pada Analisis Sentimen Twitter Berbahasa Indonesia. S1. *Program Teknologi Informasi dan Ilmu Komputer,* Universitas Brawijaya.

Qiang, G., 2010. An effective algorithm for improving the performance of Naive Bayes for text classification. *2nd International Conference on Computer Research and Development, ICCRD 2010,* (1), pp.699-701.

Rega, K.P., Wijaya, H., 2014. *Aplikasi Text Mining untuk Automasi Penentuan Tren Topik Skripsi dengan Metode K-Means Clustering.* Jurnal Cybermatika. Vol. 2 No. 1.

Rungsawang, A., Tangpong, A., Laohawee, P., and Khampachua, T. (1999). Novel Query Expansion Technique using Apriori Algorithm. In: The Eighth Text Retrieval Conference (TREC 8). Gaithersburg, Maryland, 16 November 1999. United Kingdom: NIST.

Samuel, Y., Delima, R., & Rachmat, A. (2014). Implementasi Metode K-Nearest Neighbor dengan Decision Rule untuk Klasifikasi Subtopik Berita. 10, hal. 1-15.

Soucy, P. dan Mineau, G., (2005). Beyond TF IDF Weighting for Text Categorization in the Vector Space Model. In: International Joint Conference on Artificial Intelligence. Edinburgh, Scotland, UK, 2005. Denver: Professional Book Center.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., dan Demirbas, M., (2010). Short Text Classification in Twitter to Improve Information Filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Geneva, Switzerland, 19 – 23 July 2010. New York: Association for Computing Machinery.

Sulhan, Moh. (2014). *Metode Stemming Sebagai Preprocessing Pada Filter Kata Porno Melalui Aspek Pendidikan*. [online] Yogyakarta: SENTIKA 2014.

Syaifullah, M. A. (2010). Implementasi Data Mining Algoritma Apriori Pada Sistem Penjualan. [e-journal]. Tersedia melalui Vokasi Universitas Halu Oleo.

Zafikri, A. (2008). Implementasi Metode Term Frequency Inverse Document Frequency (TF-IDF) Pada Sistem Temu Kembali Informasi. S1. *Tugas Akhir Program Studi Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara : Medan*.

Zelikovitz, S. and Marquez, F. (2005). Transductive Learning For Short-Text Classification Problems Using Latent Semantic Indexing. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02), pp.143-163.

DAFTAR LAMPIRAN

A. Data Latih

No.	Data *tweets* detik.com & kompas.com
1.	Milo Football Championship: Mencari Bibit Sepakbola dari Jenjang SD
2.	Mengapa Vendor Elektronik Jepang Tak Sejaya Dulu?
3.	Riset RSP Persahabatan: Sopir Gemuk Rentan Kecelakaan Akibat Microsleep
4.	Kartu ATM Boleh Tanpa Chip, Saldo Maksimal Rp 5 Juta
5.	Kartu ATM Pakai Chip, Tarik Tunai Bisa Rp 15 Juta/Hari
6.	Mitsubishi Enggan Beri Diskon untuk Mobil Baru.
7.	BMW R 1200 GS Laris di Indonesia
8.	BMW Siapkan X1 Tujuh Kursi Terbaru
9.	#MostPopular Teknik mesin Universitas Trisakti Gelar Pameran Otomotif
10.	Nissan dan Mitsubishi Kerjasama Buat Mobil Mungil dan Mobil Listrik
11.	Datang Akhir 2016, Maserati Australia Berharap SUV Levante Pacu Penjualan
12.	Atasi Skandal Emisi, VW Perluas Teknologi Mobil Listrik
13.	November 2015, Ducati Siap Kenalkan Motor Bermesin Kecil?
14.	Mobil-Mobil yang Membuat Wanita Terpesona
15.	Rupiah Menguat, Permintaan Mobil Baru Bakal Naik
16.	Audi TT RS Terbaru Bakal Punya Tenaga 400 Daya Kuda
17.	Prestige Belum Berencana Boyong Tesla Model X ke Indonesia
18.	Mesin Bermasalah, Pemilik Chevrolet Corvette Gugat General Motors
19.	Instruktur Keselamatan Bermotor Indonesia Cetak Prestasi di Jepang
20.	Kredit Mobil Via Online, Minta Diskon Masih Bisa Kok
21.	Intip Spek Tunggangan Kaisar Jepang
22.	Wih, Yamaha Bikin Mobil Lagi
23.	Begini Cara Piaggio Indonesia Besarkan Nama Moto Guzzi
24.	TACI Adakan Kejurda Time Rally dan Resmikan Chapter Wajo
25.	Ada Kredit Online, Matikan Salesman?
26.	#MostPopular 'Makin Banyak Mobil Mewah di Jalanan, Makin Makmur Negara'
27.	Mobil Super Mewah Toyota Century
28.	Bagi Perusahaan Pembiayaan, LCGC Jadi Primadona
29.	'Makin Banyak Mobil Mewah di Jalanan, Makin Makmur Negara'
30.	Beli Aprilia di Importir Umum Bisa Servis di Bengkel Resmi Piaggio

31.	RS Persahabatan Resmikan Layanan Kemoterapi One Day Care
32.	Apakah Bisa Hamil Meski Saat Berhubungan Tak Selalu Mencapai Orgasme?
33.	Kanker Paru Sering Tidak Bergejala, Jangan Malas Periksa Paru
34.	Adakah Prosedur Bedah Khusus untuk Memperbesar Puting?
35.	Jutaan Orang di AS Dilabeli Gemuk Meski Kenyataannya Sehat
36.	Suhu Makin Panas, Nyamuk Aedes Bisa Lebih Mudah Menyebarkan Penyakit
37.	Gusi Berdarah Setelah Menyikat Gigi? Bisa Jadi Gejala Gingivitis
38.	Idap Narkolepsi Setelah Vaksin Flu Babi, Bocah Ini Dapat Ganti Rugi Rp 2,3 M
39.	Catat! Ini Pesan Dokter Jika Hendak Jalani Operasi Pengecilan Payudara
40.	Hair Tourniquet, Ketika Sehelai Rambut Membuat Bayi Terancam Amputasi
41.	Kenali, Begini Tahapan Operasi Pengecilan Payudara
42.	Ibu 60 Tahun Ini Berjuang Agar Bisa Hamil dari Sel Telur 'Peninggalan' Anaknya
43.	Breast Reduction Jadi Solusi Utama Atasi Payudara yang Terlalu Besar
44.	Studi: Sakit Jantung Pada Survivor Kanker Sering Kali Lebih Membahayakan
45.	Istri Sering Gatal-gatal Semenjak Suntik KB, Mengapa?
46.	Merinding, Lebih dari 1.000 Semut Besar Bersarang di Kuping Bocah Ini
47.	Macromastia, Ketika Payudara Berukuran Besar Tanpa Operasi
48.	Ketika Seseorang Bertahun-tahun Tak Keluar Rumah, Begini Dampaknya
49.	Takut Kena Penyakit karena Gemuk, Maharani Pangkas Bobot 24 Kg
50.	Negara-negara Ini Masuk dalam Travel Advisory Menkes Terkait Zika
51.	Dianggap Lebih Empati, Wanita Lebih Mudah 'Tertular' Menguap
52.	Cegah Obesitas, Pilih Menu Tinggi Protein untuk Sarapan Anak
53.	Masalah Baru Setelah Harga Bensin Premium dan Solar Turun
54.	Wanita Ini Cari Pengobatan untuk Kembalikan Wajahnya yang 'Runtuh' Separuh
55.	Biopsi Malah Bikin Sel Kanker Menyebar? Itu Mitos
56.	Mazda Produksi CX-3 di Thailand
57.	Mercedes-Benz Kenalkan MPV Otonom Besok
58.	Ini SUV Land Rover Discovery Teranyar
59.	Nissan Bakal Pamerkan Konsep Mobil Listrik Otonom
60.	Mesin VTEC Terbaru Honda Bakal Lebih Mantap dengan Turbocharger
61.	Idap Narkolepsi Setelah Vaksin Flu Babi, Bocah Ini Dapat Ganti Rugi Rp 2,3 M
62.	Ini Sebabnya Mengapa Hidung 'Meler' Setelah Anda Makan Makanan Pedas
63.	Selain Urip, Ini Orang-orang yang Hidup dengan Neurofibromatosis

64.	Ini Hubungannya Kesehatan Otak dengan Kehidupan Seksual
65.	'Sakuri' Cara Sederhana dan Murah Meriah untuk Deteksi Dini Kanker Kulit
66.	Google Lepas Blokir Aplikasi Anti Iklan Samsung
67.	Alex, Karakter Pertama DLC Street Fighter V
68.	Soal Stiker LGBT Line, DPR akan Panggil Menkominfo
69.	Apple bakal Dituntut Karena 'Error 53'
70.	Mengintip Peluang IoT dari Truk Sampah
71.	Line Indonesia Tarik Semua Stiker LGBT
72.	Snapdragon 820 vs Apple A9, Siapa Menang?
73.	Kisah Fotografer Remaja Keliling 25 Negara dengan Kapal
74.	Twitter Melorot, Vine Ikut Merosot
75.	Tewas Dibunuh Setelah Memasang Iklan Mobil di Internet
76.	Dilema Saat Anak Merengek Minta Smartphone
77.	Keindahan Musim Dingin di Belarusia
78.	Vendor Android Mulai Lirik Ubuntu
79.	Stiker Gay di Line Bikin Geger
80.	Samsung Gear VR: Tenggelam ke Dunia Virtual
81.	'Kera Sakti' Coba Wujudkan Sinergi Operator dan OTT
82.	Hadir di Android dan iOS, Final Fantasy 9 Kuras 8 GB
83.	Tips Internet Aman untuk Si Kecil
84.	Portal Lamudi Disuntik Rp 442 Miliar
85.	Twitter yang Semakin Tak Berharga
86.	Kantor Digerebek, Sindikat Dedemit Maya Mati Kutu
87.	#MostPopular Berganti Akun Instagram Sekarang Lebih Gampang
88.	Helion Pesaing Balon Google Mengudara di Bandung Pekan Ini
89.	Menanti Akhir Suara Pelanggan 'Kembalikan IndiHome'
90.	Galaxy S7 Menggema, S6 Pangkas Harga
91.	Pengguna Apple Watch Jadi VIP di Fashion Show
92.	Urusan keamanan internet dan virus hingga cara aman bertransaksi online, ada
93.	Soal gadget, ada @gadtorade yang jago ngomongin semua hal mulai dari lupa password sampai melacak ponsel yang hilang
94.	Barca Cuek Soal Rekor 29 Laga Beruntun Tak Kalah
95.	Meski Ditawari Gaji Selangit, Turan Tak Tertarik Main di China
96.	Valdes Curhat, Sempat Merasa Sangat Kesepian di MU

97.	Klopp: Semoga Tak Ada Lagi Pemain yang Cedera
98.	Pacar Pato Sibuk Cari Rumah di London
99.	Untuk Harga Diri Neville dan Valencia
100.	Kapan Bikin Gol Lagi, Benteke?
101.	City yang Tak Berdaya Melawan Tim-tim Enam Besar
102.	Atletico Taklukkan Eibar 3-1
103.	Taklukkan Watford, Spurs Geser City dari Posisi Kedua
104.	Buang Keunggulan Dua Gol, Liverpool Diimbangi Sunderland 2-2
105.	Lallana Tambah Keunggulan Liverpool
106.	Firmino Bawa Liverpool Unggul 1-0
107.	City Akui Leicester Main Lebih Baik
108.	Aspac Dikejutkan Hangtuah, Pelita Jaya Dikalahkan CLS Knights Lewat OT
109.	Leicester Pecundangi City 3-1
110.	FT: City 1-3 Leicester
111.	'90 Empat menit injury time. City masih tertinggal 1-3 dari Leicester
112.	'86 Aguero memperkecil kedudukan dengan gol melalui kepalanya. City 1-3
113.	Syal Guardiola Sudah Dijual di Etihad Stadium
114.	Huth Bikin Gol Lagi, Leicester Unggul 3-0
115.	'68 Vardy nyaris mengubah skor jadi 4-0. Tendangannya gagal melewati Hart dalam posisi satu lawan satu
116.	Dari tendangan sudut tandukan Huth melambung ke tiang jauh dan masuk ke gawang Hart. City 0-3 Leicester.
117.	Sakit, Klopp Absen Dampingi Liverpool Lawan Sunderland
118.	Kepengurusan INASGOC Direvisi, Pekerjaan Panpel Belum Maksimal
119.	Barca Bukan Klub Terakhir Mascherano
120.	Mencari Penerus Chris John dan Daud Yordan
121.	Merasa Bersalah, Hazard Minta Maaf pada Mourinho
122.	Hasil Pertemuan Kepala SKK Migas dan Menperin Soal Kilang LNG Masela
123.	Kapan Tambah Daya Listrik Gratis? PLN: Mudah-mudahan Bulan Ini
124.	Ini Alasan PLN Tolak Harga Uap Panas Bumi Yang Dijual Pertamina
125.	PTDI Produksi 50 Unit Jet Tempur Made in Bandung
126.	6 Ruas Tol Trans Sumatera Masuk Tahap Konstruksi
127.	Bahas Kilang LNG Masela, Kepala SKK Migas Sambangi Saleh Husin
128.	Pertamina-PLN Berselisih, Darmin Hingga Rini Merapat ke Kantor JK

129.	BTN Bentuk Perusahaan Asuransi Jiwa
130.	I Wayan Agus Gantikan Chandra Hamzah Jadi Komut BTN
131.	Bangun Pabrik Mobil di RI, Investor China Gelontorkan Rp 5 Triliun
132.	Bikin Jet Tempur Bersama, RI dan Korsel Guyur Rp 111,5 T
133.	Garap Proyek 35.000 MW, PLN Dapat Pengawalan dari Kejagung
134.	Proyek Tol Kalimantan dan Sulawesi di Era Jokowi
135.	Proyek Tol Pertama di Kalimantan dan Sulawesi
136.	Dolar AS Tiba-tiba Jatuh ke Rp 13.870
137.	Soal Harga Uap, Kementerian BUMN Akan Pertemukan PLN dan Pertamina
138.	Kendala Proyek Listrik di RI: Perizinan Lelet dan Pembebasan Lahan
139.	Pabrik Mobil China Pertama di RI Mulai Dibangun, Ini Penampakannya
140.	Harga Minyak Turun Tajam ke Bawah US$ 35/Barel
141.	Kiwoom Securities: IHSG Bakal Cenderung Melemah
142.	Penjelasan Pertamina Soal Negosiasi Panas Bumi dengan PLN
143.	Pertamina dan PLN Belum Bersepakat Soal Harga Panas Bumi
144.	Harga Minyak Jatuh Lagi ke Titik Terendah
145.	Anjlok 7%, Bursa Saham China Disuspen (Lagi)
146.	Dolar AS Tiba-tiba Jatuh ke Rp 13.870
147.	Soal Harga Uap, Kementerian BUMN Akan Pertemukan PLN dan Pertamina
148.	Kendala Proyek Listrik di RI: Perizinan Lelet dan Pembebasan Lahan
149.	Penjelasan Baru Pertamina Soal Negosiasi Harga Uap Dengan PLN
150.	Pabrik Mobil China Pertama di RI Mulai Dibangun, Ini Penampakannya
151.	Emas Antam Naik Jadi Rp 547.000/Gram
152.	PLN: Elektrifikasi RI Kalah Dibanding Malaysia dan Thailand
153.	Ketika Ellyas Pical Kembali Naik Ring
154.	Daud Yordan Pertahankan Gelar
155.	Selesai Jalani Operasi, Caceres Absen sampai Akhir Musim
156.	Menang Angka atas Kato, Daud Yordan Pertahankan Gelar
157.	Satria Muda Tutup Seri II dengan Kemenangan
158.	Soal Tren Negatif Arsenal, Wenger: Tak Ada Alasan untuk Panik
159.	Terkait Bayern dan City, Guardiola Ibaratkan dirinya seperti Perempuan
160.	Sacchi: Sarri Poles Pertahanan Napoli, Juve Sudah Benar-benar 'Bangun'
161.	Manokwari Valeria Tak Ada Kabar, Proliga Putri Diramaikan 5 Tim

162.	Falcao: Bagaimana Mau Main, Berlatih Saja Sulit
163.	LG Luncurkan Ray, Ponsel Gahar Tak Harus Mahal
164.	Cara Mudah Merawat Baterai Ponsel Tetap Awet
165.	Xiaomi Diburu Hipster, BlackBerry Dicari Oldies
166.	Samsung Bikin Chip untuk Mobil Audi
167.	Lazada 'Jewer' Penjual Ponsel BM
168.	Tips Memilih Headphone dari Sennheiser
169.	Huawei Siapkan Pesaing Redmi Note 3
170.	Mito Terjun ke Pasar Action Cam Murah
171.	TripAdvisor Yelp Protes Keras Pencarian Google
172.	Ponsel China Ini Punya 2 Baterai, Kuat 4 Hari
173.	Keren! Sepatu dengan Wallpaper yang Bisa Gonta-ganti
174.	PLN Tak Masalah Bila Pertamina Hentikan Pasokan Uap PLTP Kamojang
175.	Ini Cara Suzuki Jaring Calon Pebalap MotoGP di Indonesia
176.	Laris di Inggris, Penjualan Global MV Agusta Naik 30 Persen
177.	Shelby Cobra 1964 Ini Diperkirakan Laku Rp 46,2 Miliar
178.	Kini, Ford Ranger Jadi Pikap Terlaris di Eropa
179.	BMW Mulai Promosikan Motor Hasil Kerjasama dengan TVS
180.	Nissan Perkirakan Penjualan Mobil Tahun 2016 Masih Lesu
181.	Ramusa, Supercar Berkarakter SUV yang Tangguh di Medan Off-Road
182.	#MostPopular YBBA Gelar Kontes Mekanik
183.	Ini Keuntungan Beli Chevrolet Orlando di Desember 2015
184.	Yayasan Dharma Bhakti Astra Gelar Kontes Mekanik Bengkel Binaan
185.	Perlu Dikenali, Apa Sih Pradiabetes Itu?
186.	Bukan Ngidam Tak Keturutan, Ini Penyebab Ngiler yang Sebenarnya
187.	Pada Anak, Kebiasaan Gigit Bibir Bisa Picu Benjolan Seperti Ini
188.	Studi: Utang Orang Tua Pengaruhi Keseimbangan Emosi Anak
189.	Gigi Susu Kan Pasti Tanggal, Kenapa Harus Disambung Kalau Patah?
190.	Sering Stres Saat Muda, Awas Risiko Diabetes Mengancam Ketika Tua
191.	Catat, Ini Saran Dokter Saat Gigi Anak Patah
192.	Tulang Kering Kanan Nyeri karena Terbentur Separator Busway
193.	Di Mobil Ini, Seseorang Bisa Merasakan Apa yang Dialami Pasien Demensia
194.	Waspada Virus Zika, El Savador Sarankan Tunda Hamil Selama 2 Tahun

195.	Garuda Tambah 23 Pesawat Baru di 2016
196.	Ayo Pakai Pertamax, Harganya Turun Mulai 5 Januari
197.	Produksi Gula PTPN X Capai 431.000 Ton dalam Setahun
198.	Awal Tahun, Harga Cabai Merah di Jakarta Tembus Rp 50.000/Kg
199.	Mascherano: Persaingan di La Liga Akan Ketat sampai Akhir
200.	Kunjungi Priok, Menhub Ingin Layanan Ekspor-Impor Buka 7x24 Jam

B. Data Uji

No.	Data *tweets* detik.com & kompas.com
1.	Jutaan Orang di AS Dilabeli Gemuk Meski Kenyataannya Sehat
2.	SUV Pertama Maserati Levante Diuji di Wilayah Dingin
3.	Seluruh Kartu ATM Harus Pakai Chip di 2022, Ini Tahapannya
4.	Akhirnya Bikin Gol Juga, Hazard
5.	Lorenzo: Wajar Kalau Yamaha Lebih Suka Rossi yang Juara
6.	Tahun Ini Oezil Belum Tambah Assist, tapi Sudah Cetak 1 Gol
7.	Lempar Bola Tenis: Bentuk Protes Fans Dortmund akan Mahalnya Harga Tiket
8.	Kalahkan Leverkusen, Bremen Melaju ke Semifinal
9.	Barca Harus Lupakan Keunggulan 7 Gol
10.	Neville Fokus Bangkitkan Valencia, Tak Mau Bahas Masa
11.	Enrique: Guardiola Bukan Ancaman bagi Barcelona
12.	Buffon: Lawan Bayern, Peluang Juve Tipis
13.	City Takluk dari Leicester, Aguero: Tak Ada Alasan Bisa Kalah
14.	Semakin Lemah, Dolar AS Sempat Sentuh Rp 13.700
15.	Ditugasi Impor Gula dan Jagung, Bulog Butuh Pinjaman Rp 3,5 T
16.	Wall Street Makin Jeblok, Jatuh Hingga 3%
17.	Jaga Yuan, Cadangan Devisa Bank Sentral China Turun US$ 500 Miliar di 2015
18.	BTN Salurkan Kredit untuk 438.000 Rumah, Mayoritas KPR Subsidi
19.	Percepat Proyek Listrik, Bank BUMN Diminta Bantu PLN
20.	Harga Premium, Pertamax Hingga Bensin Shell Turun
21.	Menperin: Jangan Jadikan Energi Sebagai Komoditi
22.	65% KPR Subsidi BTN Diserap Nasabah di Pulau Jawa
23.	Portal Lamudi Disuntik Rp 442 Miliar
24.	Twitter yang Semakin Tak Berharga
25.	Kantor Digerebek, Sindikat Dedemit Maya Mati Kutu
26.	Helion Pesaing Balon Google Mengudara di Bandung Pekan Ini
27.	Menanti Akhir Suara Pelanggan 'Kembalikan IndiHome'
28.	Galaxy S7 Menggema, S6 Pangkas Harga
29.	Pengguna Apple Watch Jadi VIP di Fashion Show
30.	Urusan keamanan internet dan virus hingga cara aman

31.	Soal Gadget, ada @gadtorade yang jago ngomongin semua hal, mulai dari lupa password sampai melacak ponsel yang hilang
32.	Untuk Fotografi, kita ada Penulis dari Info Fotografi @enchezein yang ahli
33.	Alfa Romeo Bantah Penjualan Sedan Guilia Tertunda Karena
34.	Mobil Listrik Bentley Siap Diramu
35.	Vespa dan Piaggio Naik Harga Rp 1-2 Juta
36.	Subaru XV Crossover Siap Ganti Tampang
37.	Harley Asia Pasifik Cari Diler Pengganti Mabua
38.	Mesin Satria Bakal Dipasang Suzuki di Model Lain, Bisa Skutik atau Sport
39.	Begini Tampilan Varian SUV Mercedes-Benz GLA 180 Termurah
40.	Toyota Masih Andalkan Avanza Tahun 2016
41.	Untuk Pertama Kalinya Sejak 2011, Ford Berjaya di Eropa
42.	Selain Urip, Ini Orang-orang yang Hidup dengan Neurofibromatosis
43.	'Sakuri' Cara Sederhana dan Murah Meriah untuk Deteksi Dini Kanker Kulit
44.	Beragam Gejala Baru yang Timbul Pada Survivor Ebola
45.	Tulisan 'Imitasi' di Label Pangan Disebut Bisa Jadi Solusi Masalah Obesitas
46.	Konsumsi Herba Bisa Bantu Sembuhkan Kanker? Begini Tanggapan Dokter
47.	Bocah Ini Meninggal Setelah 35 Kali Dokter Abaikan Gejala Kanker Darahnya
48.	Soal Kabar Penularan Zika Melalui Hubungan Seks, Ini Kata Menkes
49.	Memulihkan Lutut yang Kaku Pasca operasi
50.	Tumor Otak Gadis Ini Terdeteksi dari Senyumnya yang Asimetris